

**Perpustakaan SKTM**

**UNIVERSITI MALAYA**

---

**TRANSLATION TOOL FOR  
ENGLISH/MALAY  
AND MALAY/ENGLISH DOCUMENT**

**OLEH**

**MOHD IDRIS HJ ODDING  
NO. MATRIKS WEK000462**

**DI BAWAH PENYELIA  
PUAN NORISMA IDRIS**

**DAN MODERATOR  
DR. RUKAINI HJ ABDULLAH**

**DIHANTAR KEPADA  
FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT  
UNIVERSITI MALAYA**

**UNTUK MEMENUHI KEPERLUAN BAGI  
IJAZAH PERTAMA  
SAINS KOMPUTER  
SESI 2002/2003  
TARIKH PENGHANTARAN (18 FEBRUARI 2003)**

## ABSTRAK

Alat penterjemah merupakan suatu proses yang menggunakan perisian komputer yang telah dibangunkan untuk menterjemah suatu teks atau dokumen kepada bahasa tabii yang lain. Sistem yang dibangunkan ini merupakan alat penterjemah daripada Bahasa Melayu ke Bahasa Inggeris atau sebaliknya. Alat ini berfungsi untuk menterjemah suatu dokumen sama ada dalam bentuk teks atau emel dan sebagainya. Pemprosesan bahasa tabii (NLP) merupakan sumber utama dalam membangunkan sistem ini. Perisian Visual Prolog 5.2 digunakan sepanjang pelaksanaan rekabentuk dan pengimplementasian projek ini. Metodologi prototaip digunakan bagi mengawal dan mengurus pembangunan sistem ini. Pada hakikatnya, alat penterjemah ini bukan suatu proses yang mudah dengan hanya menterjemah suatu teks daripada suatu perkataan kepada perkataan yang lain. Ia sebenarnya melibatkan proses menganalisis suatu ayat atau frasa berdasarkan peraturan-peraturan bahasa dan seterusnya menterjemah perkataan dan frasa tersebut ke dalam konteks asal suatu dokumen. Sistem ini dibangunkan menggunakan pendekatan *Rule-Based System* di mana ia lebih kepada teknik penyepadanan corak dengan bantuan rantai ke hadapan dan rantai ke belakang.

Matlamat utama projek ini adalah untuk menyediakan suatu fungsi penukaran Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya berdasarkan objektif seperti memperbaiki sistem yang sedia ada, membolehkan penterjemahan dua hala (BM<->BI) dengan menggunakan komputer, boleh dijadikan sebagai kamus dwibahasa, meningkatkan atau menggalakkan lagi penggunaan Bahasa Melayu dan Bahasa Inggeris. Selain itu, dapat membangunkan satu aplikasi yang menggunakan teknik-teknik utama di



dalam kepintaran buatan. Analisa sistem sedia ada seperti EWGate's EWTranslate, Penterjemah epedoman dan Penterjemah Inggeris-Melayu dijalankan untuk mengkaji sejauh mana keberkesanan sistem tersebut dalam mencapai objektif yang telah digariskan.

Sistem ini merangkumi fungsi-fungsi seperti pemprosesan analisis di mana ia melibatkan dua elemen utama iaitu dalam penentuan tatabahasa dan penghuraian bagi menentukan jenis kategori perkataan. Pentafsiran semantik juga merupakan aspek yang penting di mana ia menentukan makna suatu perkataan, ayat atau frasa. Proses ini penting untuk mengatasi masalah *ambiguous*. Bagi menjelaskan lagi makna suatu ayat atau frasa, kontekstual/pentafsiran pengetahuan digunakan.

Sistem penterjemah ini mempunyai had-had tertentu seperti wujudnya masalah *ambiguous* di mana suatu perkataan itu mempunyai dua atau lebih makna yang berlainan. Selain itu, sistem ini juga tidak dapat menterjemah bahasa berirama atau lebih bersifat sastera. Sistem ini juga tidak dapat menterjemahkan dokumen atau teks yang menggunakan bahasa pasar. Oleh kerana bilangan kosa kata yang wujud dalam Bahasa Melayu dan Bahasa Inggeris adalah besar maka kemungkinan besar terdapat perkataan yang tidak dapat diterjemahkan. Selain itu, laporan ini juga telah membincangkan mengenai penukaran modul-modul dan algoritma kepada arahan yang dilaksanakan menggunakan perisian *Visual Prolog 5.2* dan strategi pengujian yang dijalankan bagi mengesan ralat yang wujud dalam sistem ini.

## PENGHARGAAN

### **Bismillahirrahmanirahim**

Pujian-pujian dan syukur ke hadrat Yang Maha Esa kerana dengan izin-Nya maka latihan ilmiah 1 ini dapat disiapkan pada masa yang telah ditetapkan.

Sekalung budi dan terima kasih kepada individu-individu yang terlibat dalam usaha untuk menyiapkan projek ini. Tanpa bantuan mereka dengan izin Allah mungkin projek ini tidak dapat disiapkan seperti yang dijangka.

Penghargaan dan penghormatan kepada penyelia, Puan Norisma Idris yang ikhlas membantu dan memberikan garis panduan dan nasihat di sepanjang penglibatan saya dalam projek ini. Tidak lupa juga kepada Dr. Rukaini yang selaku moderator saya di mana telah banyak memberi komen ke atas sistem ini dan telah mendorong saya untuk memperbaiki sistem yang dibangunkan ini.

Setulus ingatan kepada ibu bapa dan adik-beradik saya yang banyak membantu dari segi kewangan dan sokongan serta iringan doa yang berpanjangan.

Terima Kasih kepada semua yang terlibat dan jasa kalian hanya Allah dapat membalasnya.



## SENARAI ISI KANDUNGAN

ABSTRAK	I
PENGHARGAAN	III
SENARAI ISI KANDUNGAN	IV
SENARAI RAJAH	VI
SENARAI JADUAL	VII
<b>BAB 1 PENGENALAN</b>	<b>1</b>
1.1 PENGENALAN	1
1.2 DEFINASI MASALAH	1
1.3 OBJEKTIF	2
1.4 SKOP PROJEK	3
1.5 SASARAN PROJEK	4
1.6 KETERHADAN BAGI SISTEM PENTERJEMAHAN INI	4
1.7 RANCANGAN PERLAKSANAAN PROJEK	5
<b>BAB 2 KAJIAN LITERASI</b>	<b>7</b>
2.1 PENGENALAN KEPADA PEMROSESAN BAHASA TABII	7
2.2 SUMBER RUJUKAN	17
2.3 MASALAH YANG PERLU DIPERTIMBANGKAN	19
2.4 ANALISA SISTEM YANG SEDIA ADA	20
<b>BAB 3 METODOLOGI</b>	<b>26</b>
3.1 PENGENALAN	26
3.2 METODOLOGI PEMBANGUNAN SISTEM	26
3.3 KAEDAH DAN TEKNIK	31
<b>BAB 4 ANALISA DAN REKABENTUK SISTEM</b>	<b>33</b>
4.1 PENGENALAN	33
4.2 KEPERLUAN FUNGSIAN	34
4.3 KEPERLUAN SPESIFIKASI	35
4.4 KEPERLUAN BUKAN FUNGSIAN	35
4.5 KEPERLUAN TEKNIKAL	37
4.6 REKABENTUK STRUKTUR SISTEM	41
4.7 ALIRAN MAKLUMAT	43
4.7 REKABENTUK ANTARAMUKA PENGGUNA	43

4.8 REKABENTUK ANTARAMUKA PENGGUNA	44
<b>BAB 5 PERLAKSANAAN/PEMBANGUNAN SISTEM</b>	<b>46</b>
5.1 PENGENALAN	46
5.2 PENGKODAN	46
5.3 PERSEKITARAN PEMBANGUNAN	49
<b>BAB 6 PENGUJIAN SISTEM</b>	<b>56</b>
6.1 PENGENALAN	56
6.2 STRATEGI PENGUJIAN	56
6.3 ANALISA KEPUTUSAN UJIAN	59
<b>BAB 7 PERBINCANGAN</b>	<b>62</b>
7.1 KEPUTUSAN DAN KESIMPULAN YANG DIPEROLEH	62
7.2 KELEBIHAN SISTEM YANG DIBANGUNKAN	63
7.3 MASALAH/KELEMAHAN SISTEM YANG DIBANGUNKAN	63
7.4 PENINGKATAN YANG DIJALANKAN TERHADAP SISTEM	64
<b>APENDIKS A (PENGKODAN)</b>	<b>65</b>
<b>APENDIKS B (PANKALAN DATA DALAMAN)</b>	<b>84</b>
<b>APENDIKS C (MANUAL PENGGUNA)</b>	<b>91</b>
<b>BIBLIOGRAFI</b>	<b>94</b>



## SENARAI RAJAH

Rajah 1.0 Carta Gant Projek	6
Rajah 2.0 Perwakilan pepohon bagi ayat <i>John ate the fish</i>	10
Rajah 2.1 Pepohon yang telah ditambah dengan ciri NUMBER	13
Rajah 2.2 Contoh penterjemahan ayat Tarmizi looks like himself	21
Rajah 2.3 Contoh penterjemahan ayat nasi goreng itu dimakan oleh Ali	21
Rajah 2.4 Contoh penterjemahan ayat <i>she saw the man who sitting beside Ali</i>	22
Rajah 2.5 Contoh penterjemahan yang menghasilkan output yang kurang tepat	23
Rajah 2.6 Contoh penterjemahan ayat <i>I like to play basketball</i>	24
Rajah 2.7 Contoh penterjemahan yang menghasilkan output yang kurang tepat	25
Rajah 3.0 Model prototaip	27
Rajah 3.1 Fasa-fasa dalam model prototaip	29
Rajah 4.0 Struktur sistem penterjemah	41
Rajah 4.1 Modul-modul bagi penganalisis	41
Rajah 4.2 Aliran maklumat sistem penterjemah	43
Rajah 4.3 Antaramuka pengguna bagi sistem penterjemah	45
Rajah 5.0 Menunjukkan perlaksanaan sistem dengan lebih terperinci	49
Rajah 5.1 <i>Project Window</i>	50
Rajah 5.2 <i>The Dialog and Window Expert</i>	51
Rajah 6.0 Langkah-langkah pengujian yang dijalankan	59
Rajah 6.1 Menunjukkan kebergantungan antara masa dan saiz Mamus yang diperlukan untuk membuat penterjemahan	60

## SENARAI JADUAL

### Pengenalan

Jadual 1.0 Menunjukkan kebergantungan antara masa dan saiz kamus

60

#### 1.1 PENGENALAN

Alat penterjemah merupakan suatu proses yang menggunakan perisian komputer yang telah dibangunkan untuk menterjemah suatu teks atau dokumen kepada bahasa lain yang lain.

#### 1.2 DEFINISI MASALAH

Sistem yang akan saya bangunkan ini merupakan alat penterjemah daripada Bahasa Melayu ke Bahasa Inggeris, dan sebaliknya. Alat ini berfungsi untuk menterjemah suatu dokumen sama ada dalam bentuk teks, surat email dan sebagainya. Selain itu sistem ini bertindak sebagai alat rujukan mendedahkan kepada pengguna sedikit sebanyak mengenai morfologi dan sintaksis yang terdapat dalam bahasa Melayu. Pada hakikatnya alat penterjemah ini bukan suatu proses yang mudah dengan hanya menterjemah suatu teks daripada suatu perkataan kepada perkataan yang lain. Sebaliknya melibatkan proses menganalisis suatu ayat atau frasa berdasarkan peraturan-peraturan bahasa dan seterusnya menterjemah perkataan dan frasa tersebut ke dalam konteks asal suatu dokumen.

Pada masa alat penterjemahan, ambiguitas merupakan masalah yang sering timbul. Contohnya perkataan 'saya' mungkin boleh diartikan sebagai orang yang sedang berucap atau mungkin dalam Bahasa Inggeris. Selain itu, masalah lain seperti dari aspek kepeluasan. Contohnya 'All voted in under last week'. Tanpa pengetahuan yang



## **BAB 1**

### **PENGENALAN**

#### **1.1 PENGENALAN**

Alat penterjemah merupakan suatu proses yang menggunakan perisian komputer yang telah dibangunkan untuk menterjemah suatu teks atau dokumen kepada bahasa tabii yang lain.

#### **1.2 DEFINISI MASALAH**

Sistem yang akan saya bangunkan ini merupakan alat penterjemah daripada Bahasa Melayu ke Bahasa Inggeris atau sebaliknya. Alat ini berfungsi untuk menterjemah suatu dokumen sama ada dalam bentuk teks atau emel dan sebagainya. Selain itu sistem ini bertindak sebagai alat yang akan mendedahkan kepada pengguna sedikit sebanyak mengenai morfologi dan semantiks yang terlibat dalam bahasa tabii manusia. Pada hakikatnya, alat penterjemah ini bukan suatu proses yang mudah dengan hanya menterjemah suatu teks daripada suatu perkataan kepada perkataan yang lain. Ia sebenarnya melibatkan proses menganalisis suatu ayat atau frasa berdasarkan peraturan-peraturan bahasa dan seterusnya menterjemah perkataan dan frasa tersebut ke dalam konteks asal suatu dokumen.

Pada suatu alat penterjemahan, ambiguous merupakan masalah yang sering timbul. Contohnya perkataan *saw*, mungkin boleh diertikan sebagai gergaji, menggergaji atau melihat dalam Bahasa Melayu. Selain itu, masalah lain seperti dari aspek kepunyaan. Contohnya 'Ali visited his mother last week'. Terdapat sistem yang

sedia ada akan menterjemahkan ayat tersebut sebagai 'Ali melawat dia emak lepas minggu'. Tetapi, terjemahan sebenar adalah 'Ali melawat emak dia minggu lepas'. Oleh itu usaha untuk membangunkan sistem penterjemahan ini adalah perlu untuk memperbaiki sistem yang sedia ada.

### 1.3 OBJEKTIF

Matlamat utama projek ini adalah untuk menyediakan suatu fungsi penukaran Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya berdasarkan objektif berikut:-

1. Memperbaiki kualiti perisian yang sedia ada seperti masalah *ambiguous* dan kepunyaan yang telah dinyatakan di atas.
2. Memberi kemudahan kepada pengguna untuk memaparkan penukaran perkataan secara dua hala, dengan menggunakan komputer yang sesuai dengan zaman teknologi maklumat.
3. Menggalakkan penggunaan dan pembelajaran Bahasa Melayu dan Bahasa Inggeris bagi meningkatkan kemahiran mereka dalam Bahasa Melayu dan Bahasa Inggeris.
4. Boleh juga dijadikan sebagai kamus perkataan dwibahasa
5. Membangunkan satu aplikasi yang menggunakan teknik-teknik utama di dalam kepintaran buatan.



## 1.4 SKOP PROJEK

Projek ini merangkumi skop-skop seperti berikut:-

### 1. Pemprosesan Kategori Perkataan

Melibatkan proses penghuraian (split), di mana struktur suatu ayat dianalisis. Proses penghuraian ini sebenarnya berfungsi untuk menentukan kategori setiap perkataan daripada suatu ayat. Contoh kategori adalah seperti kata kerja (verb), kata nama (noun) dan banyak lagi. Penggabungan major linguistik berkaitan seperti subjek-kata kerja, kata kerja-objek, dan kata nama-pengubah, proses penghuraian itu dapat membekalkan rangka kerja untuk proses pentafsiran semantiks.

### 2. Proses Pentafsiran Semantik

Ia melibatkan proses mewakili makna suatu teks. Makna suatu teks itu diwakilkan dalam bentuk logikal (logical form). Proses pentafsiran semantiks ini akan menggunakan pengetahuan makna sesuatu perkataan dan struktur linguistik seperti peranan yang dimainkan oleh kata nama atau ketransitiviti suatu kata kerja seperti suatu perkataan mungkin mempunyai beberapa makna yang berbeza. Proses ini adalah penting dalam menentukan penggunaan tatabahasa dan struktur yang betul.

### 3. Kontekstual/Pentafsiran Pengetahuan (World Knowledge)

Proses ini akan mengembangkan makna suatu ayat itu dengan mengaitkan dunia sebenar. Proses ini bertujuan memudahkan atau melengkapkan lagi pemahaman kita terhadap suatu perkataan atau frasa.

## 1.5 SASARAN PROJEK

Secara dasarnya, sasaran projek ini adalah khusus kepada pengguna yang memiliki pengetahuan dalam Bahasa Melayu dan Bahasa Inggeris seperti para pelajar sama ada sekolah rendah, sekolah menengah atau pusat pengajian tinggi dan pengguna-pengguna lain.

Sistem ini hanya dapat menterjemah teks yang mudah sahaja dan tidak kompleks sama ada Bahasa Melayu ke Bahasa Inggeris dan sebaliknya. Sistem ini boleh digunakan sebagai kamus dwibahasa. Selain itu, ia juga boleh digunakan untuk menterjemah teks-teks yang tidak kompleks seperti dokumen dan emel.

## 1.6 KETERHADAN BAGI SISTEM PENTERJEMAHAN INI

Sistem penterjemahan yang dibangunkan ini mungkin mempunyai had-had seperti di bawah:-

- **Penentuan Ambiguous**

Sekiranya wujudnya *ambiguous*, ada dua kemungkinan iaitu sama ada sistem ini dapat ditentukan atau pun tidak. Sebenarnya ia bergantung kepada program yang telah ditetapkan.

- **Bahasa Berirama/Peringkat Tinggi**

Sistem ini tidak dapat menterjemahkan bahasa yang berirama seperti puisi-puisi, kesimpulan bahasa dan bahasa yang bersifat sastera.



- **Kekurangan Leksikon**

Berkemungkinan terdapat perkataan yang tidak dapat diterjemahkan, ini disebabkan perkataan tersebut tidak terdapat atau belum dimasukkan ke dalam pangkalan data.

- **Penggunaan Bahasa Pasar**

Sistem ini tidak dapat menterjemahkan bahasa yang menggunakan bahasa pasar.

## **1.7 RANCANGAN PERLAKSANAAN PROJEK**

Menurut perancangan, projek ini akan diselesaikan dalam jangka masa dari bulan Jun 2002 hingga bulan Feb 2003, di mana ia dibahagikan kepada dua bahagian iaitu:-

### **1. Semester 1 Sesi 2002/2003**

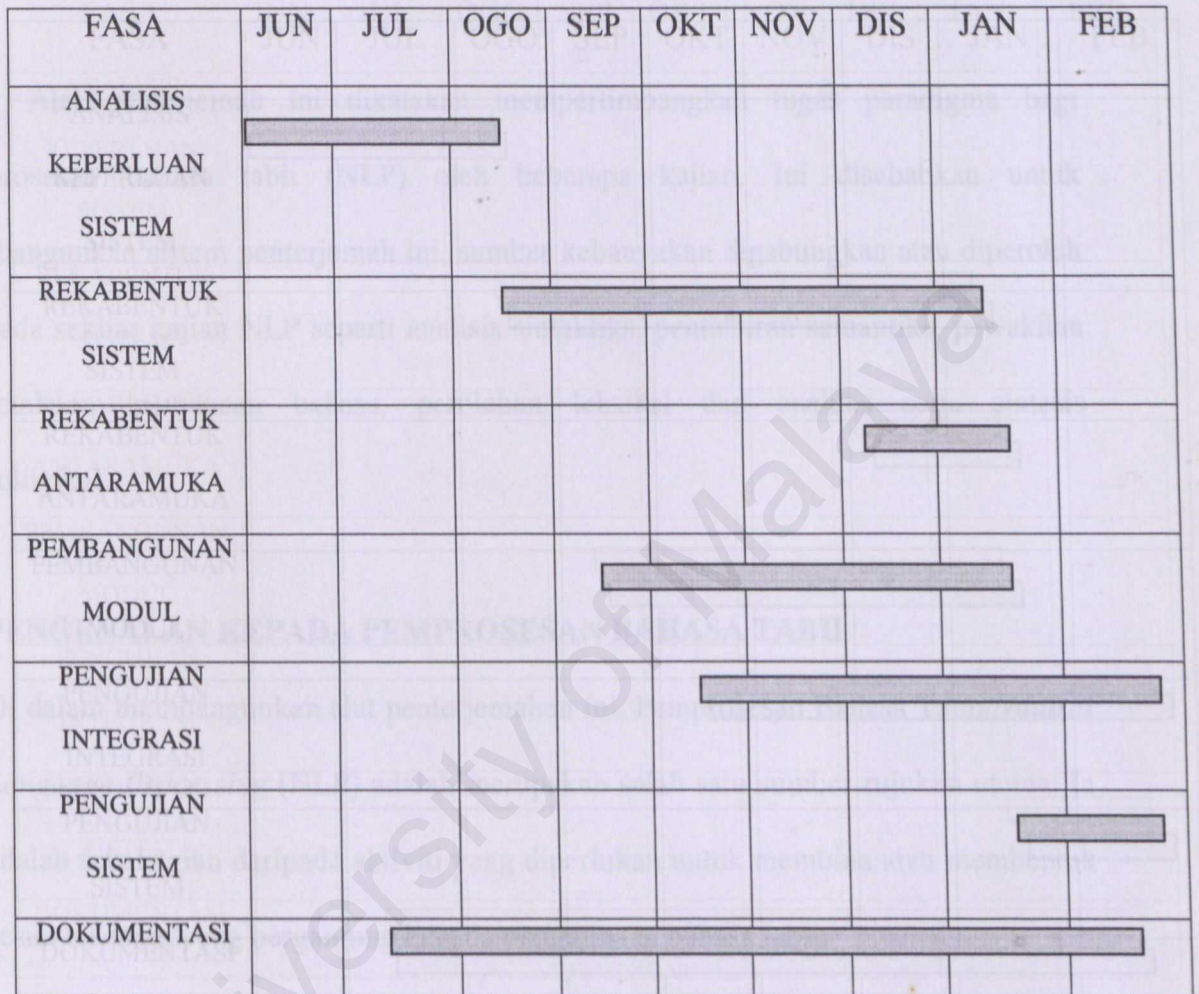
Pada peringkat ini, proses projek dan kajian mendalam berkenaan aplikasi yang ingin dibina dilakukan. Objektif dan skop aplikasi juga ditentukan. Keperluan pembangunan seperti perkakasan, perisian serta rekabentuk kasar juga ditentukan.

### **2. Semester 2 Sesi 2002/2003**

Pada peringkat ini pula, terdiri daripada pembangunan sistem di mana ia melibatkan pembangunan, pengurusan dan pengujian sistem.

## KAR 2

### KAJIAN LITERASI



**Rajah 1.0 Carta Gant Projek**



## BAB 2

### KAJIAN LITERASI

Alat penterjemah ini dikatakan mempertimbangkan tugas paradigma bagi pemprosesan bahasa tabii (NLP) oleh beberapa kajian. Ini disebabkan untuk membangunkan sistem penterjemah ini, sumber kebanyakan digabungkan atau diperoleh daripada sekitar kajian NLP seperti analisis sintaktiks, pentafsiran semantiks, perwakilan pengetahuan, penjanaan bahasa, perolehan leksikal dan analisis serta sintesis morfologikal.

#### 2.1 PENGENALAN KEPADA PEMPROSESAN BAHASA TABII

Di dalam membangunkan alat penterjemahan ini, Pemprosesan Bahasa Tabii/*Natural Language Processing* (NLP) adalah merupakan salah satu sumber rujukan utama. Ia adalah sebahagian daripada aktiviti yang diperlukan untuk membina atau membentuk sesuatu sistem yang bergantung kepada penggunaan bahasa tabii.

Objektif utama Pemprosesan Bahasa Tabii ialah bagi merekabentuk dan membina sistem komputer yang boleh menganalisa, memahami dan menjanakan bahasa tabii manusia. Sistem yang melibatkan bahasa tabii/semulajadi perlu mempertimbangkan pengetahuan mengenai struktur bahasa itu sendiri. Ia termasuklah apakah itu perkataan, bagaimana beberapa perkataan digabungkan lalu membentuk suatu ayat, apakah makna suatu perkataan, bagaimana makna suatu perkataan itu boleh membentuk suatu ayat yang bermakna dan sebagainya.

Terdapat beberapa pengetahuan yang berkaitan dengan pemahaman bahasa tabii adalah seperti berikut:-

### **1. Pengetahuan phonetik dan phonologikal**

Pengetahuan yang menunjukkan bagaimana perkataan boleh dikaitkan dengan bunyi atau lebih jelas berkenaan dengan ejaan yang mengikut sebutan.

### **2. pengetahuan morfologikal**

Pengetahuan yang menunjukkan bagaimana beberapa perkataan dapat dihasilkan daripada suatu perkataan dasar. Contohnya perkataan 'main' ditambah dengan imbuhan 'ber-' lalu menjadi 'bermain'.

### **3. Pengetahuan sintaktik**

Pengetahuan yang menunjukkan bagaimana beberapa perkataan digabungkan untuk membentuk suatu ayat atau frasa yang betul.

### **4. Pengetahuan semantik**

Pengetahuan yang menunjukkan apakah makna suatu perkataan dan bagaimana makna suatu perkataan itu boleh membentuk suatu ayat yang bermakna.

### **5. Pengetahuan pragmatik**

Pengetahuan yang menunjukkan bagaimana suatu ayat digunakan pada situasi yang berbeza dan bagaimana penggunaannya dapat mempengaruhi penafsiran suatu ayat.

### **6. Pengetahuan hujah**

Pengetahuan yang menunjukkan bagaimana ayat awalan mempengaruhi penafsiran pada ayat seterusnya.



## 7. Pengetahuan world

Pengetahuan yang melibatkan dunia fizikal contohnya interaksi sosial antara manusia. Pengetahuan ini penting untuk memahami makna sebenar pada suatu teks atau pertuturan.

### 2.1.1 Analisis Sintaktik

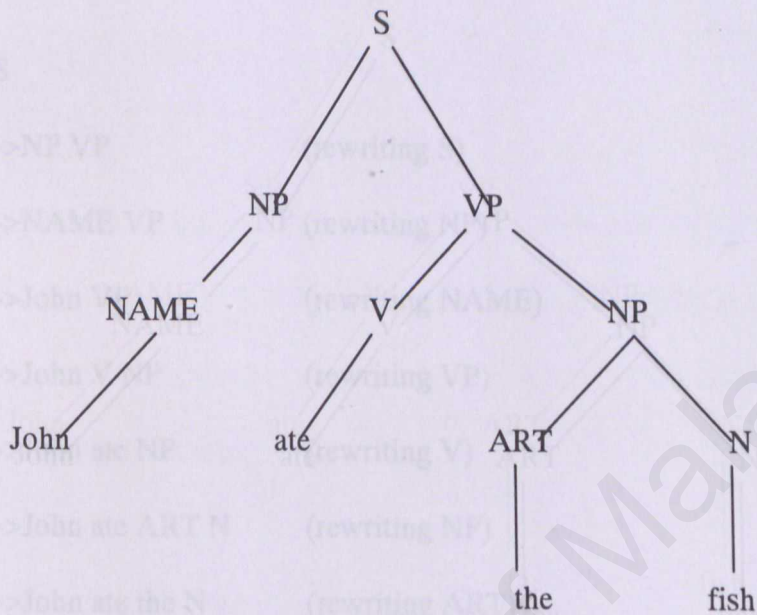
Untuk menganalisa sintaktiks struktur ayat, perlu pertimbangkan 2 elemen iaitu tatabahasa (grammar) dan teknik penghuraian (parsing technique). Pada sistem penterjemahan ini, tatabahasa adalah amat dititikberatkan. Terdapat banyak teknik penghuraian seperti penghuraian atas-bawah (a top-down parser), penghuraian bawah-atas (a bottom-up parser) dan banyak lagi. Biasanya terdapat juga program yang tidak menghasilkan huraian pepohon, ini disebabkan pepohon penghurai tidak akan dihasilkan secara luaran tetapi akan dijana dan diwakilkan secara terus pada semantik dalaman.

#### 2.1.1.1 *context free grammar* (CFG)

Biasanya sintaktiks bahasa diwakilkan dengan berdasarkan pada *context free grammar* (CFG), di mana struktur ayat diwakilkan bagaimana frasa tertentu harus ditulis dalam bentuk subfrasa yang dikandungi. Maklumat sebegini boleh diwakilkan dengan pepohon seperti Rajah di bawah. Perkataan seperti John, ate, the dan fish dikenali sebagai simbol terminal. Symbol yang lain seperti S, NP,VP dan sebagainya dikenali sebagai simbol nonterminal.

### 2.1.1.2 teknik penguraian atas-bawah

Contoh penguraian atas-bawah:



**Rajah 2.0 Perwakilan pepohon bagi ayat *John ate the fish***

Ayat bagi *John ate the fish* boleh diwakilkan dengan CFG seperti berikut:-

1.  $S \rightarrow NP VP$

2.  $VP \rightarrow V NP$

3.  $NP \rightarrow NAME$

4.  $NP \rightarrow ART N$

5.  $NAME \rightarrow John$

6.  $V \rightarrow ate$

7.  $ART \rightarrow the$

8.  $N \rightarrow fish$



### 2.1.1.2 teknik penghuraian atas-bawah

Contoh penghuraian atas-bawah:-

S

->NP VP (rewriting S)

->NAME VP (rewriting NP)

->John VP (rewriting NAME)

->John V NP (rewriting VP)

->John ate NP (rewriting V)

->John ate ART N (rewriting NP)

->John ate the N (rewriting ART)

->John ate the fish (rewriting N)

Teknik penghuraian atas-bawah ini bermula dengan simbol S  
sehinggalah terhasilnya suatu ayat yang lengkap seperti contoh di atas.

### 2.1.1.3 Teknik penghuraian bawah-atas

Contoh penghuraian bawah-atas :-

John ate the fish

->NAME ate the fish (rewriting John)

->NAME V the fish (rewriting ate)

->NAME V ART fish (rewriting the)

->NAME V ART N (rewriting fish)

->NP V ART N (rewriting NAME)

->NP V NP (rewriting ART N)

->NP VP (rewriting V NP)

->S (rewriting NP VP)

Teknik penghuraian bawah-atas ini bermula dengan perkataan suatu ayat sehinggalah terhasilnya simbol S. Jelas dapat dilihat di sini bahawa teknik penghuraian atas-bawah dengan penghuraian bawah-atas adalah saling bertentangan.

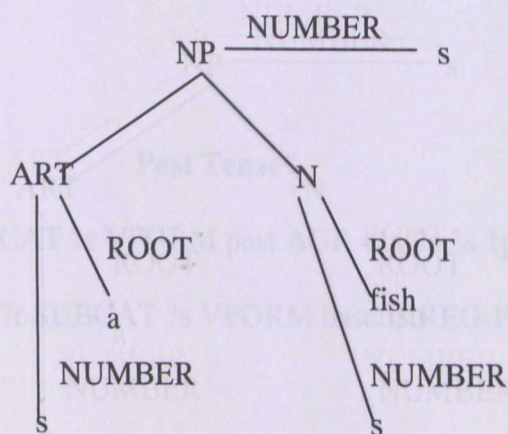
#### 2.1.1.4 Jamak (Plural) dan Tunggal (Singular)

Contohnya dalam Bahasa Inggeris, NP *a fishes* adalah tidak benar dari segi tatabahasa sebab *a* itu merujuk kepada objek yang tunggal manakala kata nama *fishes* merujuk kepada objek jamak. Fenomena ini boleh diatasi dengan menambah satu ciri iaitu NUMBER kepada CFG. Contoh adalah seperti berikut:-

NP-SING->ART-SING N-SING

NP-PLURAL->ART-PLURAL N-PLURAL





**Rajah 2.1 Pepohon yang telah ditambah dengan ciri NUMBER**

#### 2.1.1.5 Analisis Morfologikal Dan Leksikon

Pada suatu kata kerja dalam Bahasa Inggeris, ia boleh dipecahkan kepada *present tense*, *past tense*, *past participle*, *continuous tense* dan sebagainya. Semua ini boleh dikaitkan dengan morfologikal. Contohnya suatu kata dasar *play* jika ditambah dengan *s* menjadi *plays* (*present tense*), tambah dengan *ed* menjadi *played* (*past tense*) dan tambah dengan *ing* menjadi *playing* (*continuous tense*).

Leksikon terdiri daripada maklumat mengenai kesemua perkataan yang berbeza. Contohnya, bila hendak menterjemah perkataan 'mahu' daripada ayat 'saya mahu makan nasi' kepada Bahasa Inggeris. Jika diperiksa pada pangkalan data (leksikon) 'mahu' boleh diterjemah kepada *want*, *wanted*, *wanting*, dan *wants*. Masalah sekarang mana satu perkataan terjemahan yang perlu diambil. Leksikon dan morfologikal

adalah amat berkait, ia dapat mengatasi masalah tersebut seperti contoh berikut:-

### Past Tense

(V ROOT ?r SUBCAT ?s VFORM past AGR {1s 2s 3s 1p 2p 3p})->

(V ROOT ?r SUBCAT ?s VFORM base IRREG-PAST-)+ED

### Present Tense

(V ROOT ?r SUBCAT ?s VFORM pres AGR 3s )->

(V ROOT ?r SUBCAT ?s VFORM base IRREG-PAST-)+S

#### 2.1.1.6 Kata Kerja Bantu (Auxiliary Verb)

Biasanya suatu ayat Bahasa Inggeris terdiri daripada kata kerja bantu diikuti dengan kata kerja utama seperti ayat-ayat berikut:-

*I can see the house.*

*I will have seen the house.*

*I was watching the movie.*

*I should have been watching the movie.*



Ia boleh diwakilkan dengan suatu ciri misalnya COMPFORM. Contoh ditulis menggunakan LISP di mana melibatkan ciri ini adalah seperti berikut:-

```
can : ( CAT AUX  
      MODAL +  
      VFORM pres  
      AGR {1s 2s 3s 1p 2p 3p }  
      COMPFORM base)
```

```
could : ( CAT AUX  
         MODAL +  
         VFORM {pres past}  
         AGR {1s 2s 3s 1p 2p 3p }  
         COMPFORM base)
```

### 2.1.2 PENTAFSIRAN SEMANTIK

Di dalam sistem penterjemahan, ambiguiti pasti akan wujud iaitu suatu perkataan yang mempunyai dua atau lebih makna yang berbeza. Contohnya perkataan *kid*, mungkin akan wujud ambiguiti antara anak kambing dengan anak manusia. Ini boleh diatasi dengan mewujudkan dua makna iaitu BABY-GOAT1 dan BABY-HUMAN1. Makna sesuatu perkataan atau ayat boleh diwakilkan dengan bahasa bentuk logik.

Pemprosesan sintaktik dan pentafsiran semantik adalah saling bergantung antara satu sama lain. Terdapat beberapa strategi atau pendekatan yang boleh digunakan untuk membuat pentafsiran semantik. Antaranya adalah seperti berikut:-

- *Grammatical relations*
- *Semantic grammars*
- *Template matching*
- *Semantically driven parsing techniques*

#### **2.1.2.1 Contoh Pendekatan Pentafsiran Semantik menggunakan *Grammatical Relations***

Strategi atau pendekatan yang digunakan untuk membangunkan sistem penterjemah ini adalah *Grammatical Relations*. Analisis sintaktik dan pentafsiran semantik adalah saling berkait antara satu sama lain. Dengan menggunakan pendekatan ini, pentafsiran semantik dilakukan dengan mengabstrak struktur sintaktik yang telah dihuraikan pada analisis sintaktik. *Grammatical relations* sentiasa mengaitkan dengan subjek logikal (LSUBJ), objek logikal (LOBJ) dan objek tidak langsung (IOBJ). Contoh strategi grammatical relations yang menggunakan LISP adalah seperti berikut:-



### Contoh 1

Jack bought a ticket

(s1 PRED BUYS1)(s1 TNS PAST)

(s1 LSUBJ (NAME j2 "Jack"))

(s1 LOBJ <A t1 TICKET1>)

### Contoh 2

Jill gave a book to Jack

(s2 PRED GIVES1)(TNS s2 PAST)

(s2 LSUBJ (NAME j1 "Jill"))

(GIVE1 LOBJ <A b1 BOOK1>)

(GIVE1 TO (NAME j2 "Jack"))

## 2.2 SUMBER RUJUKAN

Sebelum memulakan projek ini, kajian literasi merupakan bahagian yang penting di mana melibatkan proses pengumpulan matlumat yang diperlukan untuk membangunkan alat penterjemah bahasa ini (Bahasa Melayu<->Bahasa Inggeris). Pengumpulan dibuat dengan menyeluruh dan teliti agar dapat menjadi garis panduan dalam merangka proses pembangunan aplikasi tersebut.

Proses pengumpulan ini boleh dibahagikan kepada beberapa kaedah di antaranya ialah :-

### **1. Pencarian Maklumat Melalui internet**

Sumber-sumber dari internet dilayari untuk mengumpul maklumat yang berkaitan dengan penterjemahan bahasa. Cara untuk mencari bahan rujukan ini adalah dengan menggunakan enjin pencarian seperti Yahoo, Google dan Altavista.

### **2. Bilik Dokumen**

Bilik dokumen yang terletak di Fakulti Sains Komputer dan Teknologi Maklumat digunakan untuk mengumpulkan maklumat-maklumat berlandaskan beberapa contoh laporan dari pelajar-pelajar yang lepas. Maklumat yang diperoleh ini juga banyak membantu memberikan panduan dan dorongan untuk membangunkan aplikasi ini.

### **3. Perpustakaan Dan Kedai Buku**

Selain itu, sumber rujukan juga diperoleh dengan membuat rujukan buku-buku teks, jurnal-jurnal, surat khabar, majalah dan lain-lain yang berkaitan dengan projek yang terdapat di perpustakaan utama Universiti Malaya dan juga kedai-kedai buku yang terkemuka di Kuala Lumpur seperti Kedai Buku MPH.

### **4. Berjumpa Dengan Yang Pakar**

Maklumat paling tepat diperoleh dengan menemuramah atau menghubungi orang-orang yang arif dan pakar dalam bidang ini serta bidang yang berkaitan dengan pengaturcaraan.



## **2.3 MASALAH YANG PERLU DIPERTIMBANGKAN**

### **1. Penggunaan Perisian**

Untuk menyiapkan sistem ini, perisian yang akan saya gunakan adalah Visual Prolog. Oleh kerana kurang mahir dalam menggunakan perisian Visual Prolog, ini akan memakan masa untuk saya menyiapkan projek yang telah ditugaskan. Bagi menyelesaikan masalah ini, saya cuba mendalami cara penggunaan perisian ini melalui buku dan melayar internet.

### **2. Peruntukan Masa**

Dipengaruhi oleh faktor peruntukan masa yang terhad untuk menyiapkan projek. Lebih-lebih lagi dalam proses membuat program aturcara yang memerlukan pemikiran pandang ke hadapan serta penaaakulan yang perlu dibuat bagi menghasilkan program aturcara yang baik dengan itu membolehkan sistem dapat berjalan atau berfungsi dengan baik. Selain itu, Bahasa Melayu dan Bahasa Inggeris mempunyai kosa kata yang besar dan ia memerlukan masa yang panjang untuk merekodkan kesemuanya jika dilakukan secara individu.

### **3. Ruang Pangkalan Data**

Dalam sistem ini akan menggunakan pangkalan data untuk menyimpan perkataan-perkataan Bahasa Melayu dan Bahasa Inggeris yang melibatkan sinonim, kata nama, kata kerja, kata sifat dan sebagainya. Dapat dilihat di sini bahawa sistem ini memerlukan ruang pangkalan data yang banyak.

### **4. Masalah Semantiks**

Makna suatu perkataan mungkin boleh dipecahkan kepada beberapa makna yang sama dan juga beberapa makna yang berbeza. Untuk mendapatkan suatu

penterjemahan yang baik, perlu melakukan pencarian kepada perkataan yang bersesuaian dengan ayat. Selain itu, masalah lain yang melibatkan semantiks seperti untuk mengkategorikan kepunyaan contohnya *mine, us, ours* dan lain-lain mengkategorikan kuantiti seperti *some, a lot of* dan sebagainya.

## **2.4 ANALISA SISTEM YANG SEDIA ADA**

Analisa sistem sedia ada dijalankan untuk mengkaji sejauh mana keberkesanan sistem tersebut dalam mencapai objektif yang telah digariskan. Dalam konteks ini, kajian dijalankan terhadap beberapa jenis alat penterjemah yang mempunyai ciri-ciri yang hampir sama tetapi dibangunkan dengan kaedah-kaedah yang berlainan dan mempunyai kelebihan serta kekurangan masing-masing.

### **2.4.1 Contoh Sistem Yang Sedia Ada**

#### **2.4.1.1 EWGate's EWTranslate**

EWGate's EWTranslate merupakan alat penterjemah yang dapat menterjemah Bahasa Melayu ke Bahasa Inggeris dan sebaliknya. Selain Bahasa Melayu dan Bahasa Inggeris, ia juga dapat menterjemah bahasa-bahasa lain. Pada sistem ini juga terdapat masalah mengenai penentuan kedudukan (preposition) dan kepunyaan (possession). EWGate's EWTranslate dibangunkan menggunakan teknologi pemproses bahasa tabii (NLP). Pada awal kemunculannya, ia hanya merupakan alat penterjemah yang hanya menggunakan kamus lookup yang mudah



sehinggalah sistem itu dikembangkan untuk menganalisis bahasa yang kompleks untuk melakukan penterjemahan yang cekap dan tepat.

Di bawah menunjukkan beberapa contoh penterjemahan bagi EWGate's EWTranslate:-

Please type 50 words or less.

Tarmizi looks like himself
Tarmizi serupa dengan dirinya

English -> Malay Translate

Rajah 2.2 Contoh penterjemahan ayat Tarmizi looks like himself

Please type 50 words or less.

nasi goreng itu dimakan oleh Ali
The fried rice is eaten by Ali

Malay -> English Translate

Rajah 2.3 Contoh penterjemahan ayat nasi goreng itu dimakan oleh Ali

#### 2.4.1.2 Penterjemah Epedoman

Penterjemah epedoman merupakan alat penterjemah yang dapat menterjemah Bahasa Melayu dan Bahasa Inggeris. Penterjemahannya melibatkan penterjemahan teks, penterjemahan URL dan pencarian dwibahasa.

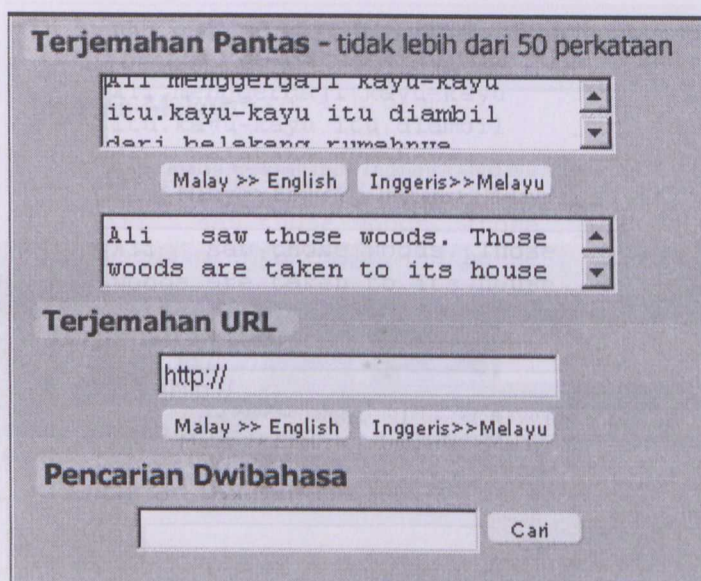
Di bawah menunjukkan beberapa contoh penterjemahan 'penterjemah epedoman':-

The screenshot displays a web-based translation tool interface. It is divided into three main sections:

- Terjemahan Pantas - tidak lebih dari 50 perkataan**: This section contains two text input fields. The first field has the text "she saw the man who sitting beside Ali". Below the fields are two buttons: "Malay >> English" and "Inggeris >> Melayu". The second field contains the Malay translation: "Dia melihat lelaki yang duduk di tepi Ali".
- Terjemahan URL**: This section has a text input field containing "http://". Below it are the same two language direction buttons: "Malay >> English" and "Inggeris >> Melayu".
- Pencarian Dwibahasa**: This section features a single text input field and a button labeled "Cari" (Search).

Rajah 2.4 Contoh penterjemahan ayat *she saw the man who sitting beside Ali*





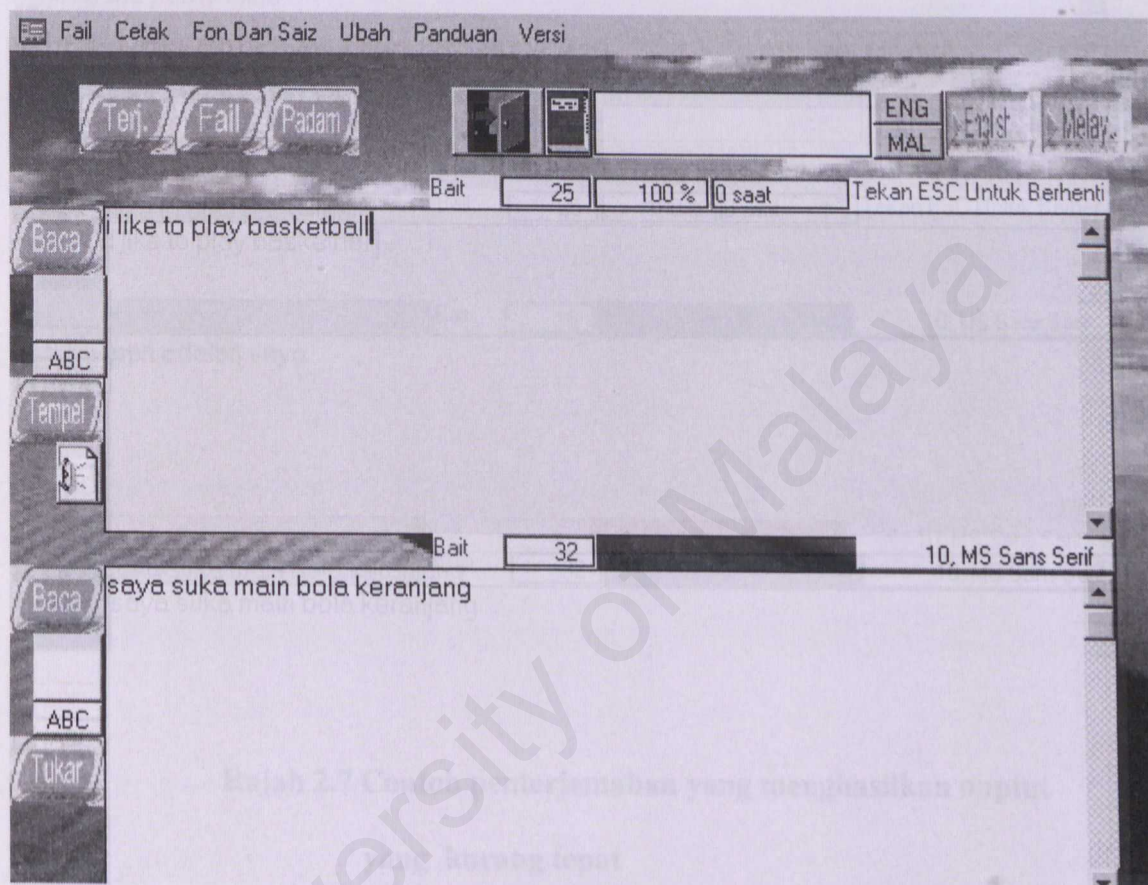
Rajah 2.5 Contoh penterjemahan yang menghasilkan output yang kurang tepat

pada contoh di atas, pada ayat 'Ali menggergaji kayu-kayu itu. Kayu-kayu itu diambil dari belakang rumahnya' diterjemahkan kepada *Ali saw those woods. Those woods are taken to its house*. Hasil penterjemahan itu kurang tepat, kita dapat lihat dengan lebih jelas pada ayat 'diambil dari belakang rumahnya' telah diterjemahkan kepada *taken to its house*. Sistem ini mempunyai masalah mengenai penentuan kedudukan (preposition) dan kepunyaan (possession).

#### 2.4.1.3 Penterjemah Inggeris-Melayu

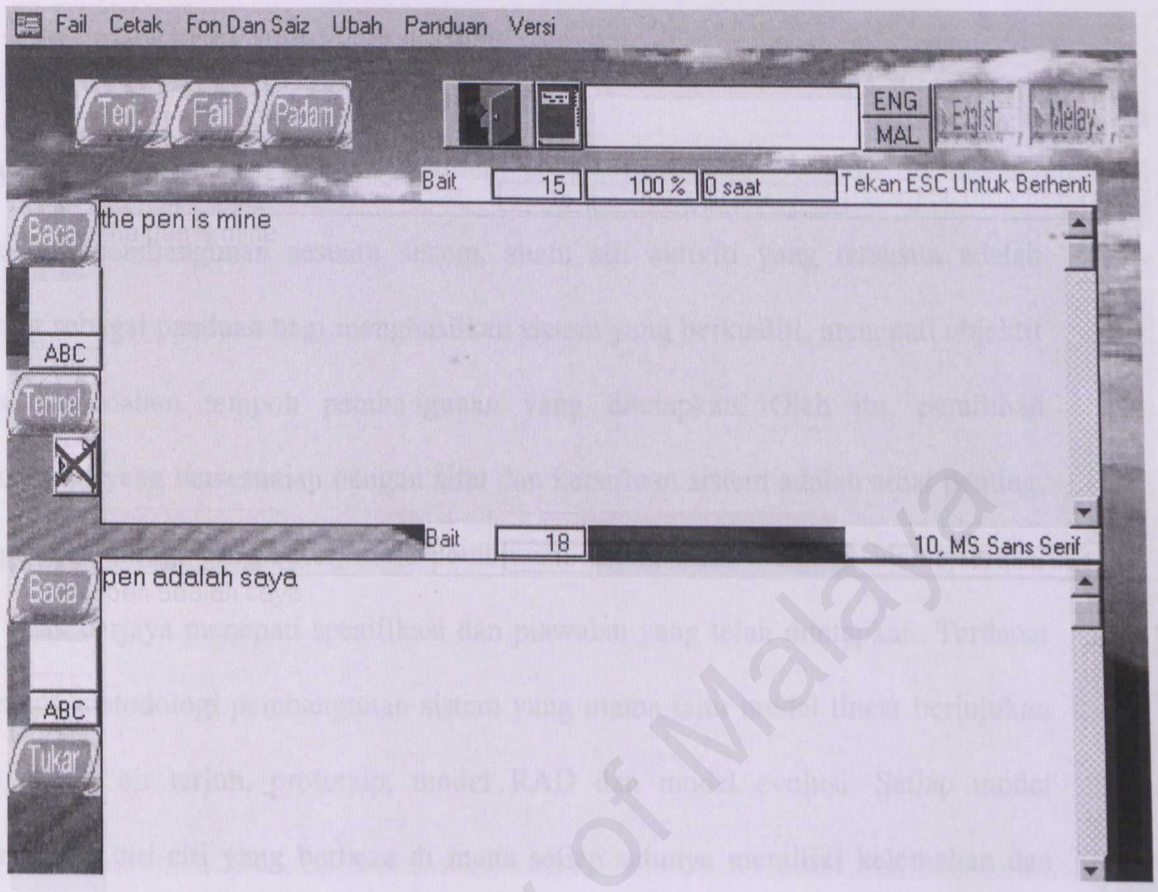
Penterjemah Inggeris-Melayu merupakan alat penterjemahan yang menterjemah Bahasa Ingeris ke Bahasa Melayu. Selain itu, perkataan yang ditulis atau yang telah diterjemahkan boleh mengeluarkan suara dengan menyebut perkataan yang dipaparkan.

Di bawah menunjukkan beberapa contoh penterjemahan 'Penterjemah  
Inggeris-Melayu':-



Rajah 2.6 Contoh penterjemahan ayat *I like to play basketball*





**Rajah 2.7 Contoh penterjemahan yang menghasilkan ouput yang kurang tepat**

pada contoh di atas, ia menghasilkan ouput yang kurang tepat di mana ayat *the pen is mine* diterjemahkan kepada 'pen adalah saya'. Sistem ini mempunyai masalah mengenai penentuan kepunyaan (possession).

## **BAB 3**

### **METODOLOGI**

#### **3.1 PENGENALAN**

Di dalam pembangunan sesuatu sistem, suatu siri aktiviti yang tersusun adalah penting sebagai panduan bagi menghasilkan sistem yang berkualiti, menepati objektif serta siap dalam tempoh pembangunan yang ditetapkan. Oleh itu, pemilihan metodologi yang bersesuaian dengan sifat dan keperluan sistem adalah amat penting. Tanpa metodologi yang betul, kerja pembinaan sistem akan menjadi tidak terkawal dan tidak berjaya menepati spesifikasi dan piawaian yang telah ditetapkan. Terdapat beberapa metodologi pembangunan sistem yang utama iaitu model linear berjujukan atau model air terjun, prototaip, model RAD dan model evolusi. Setiap model mempunyai ciri-ciri yang berbeza di mana setiap satunya memiliki kelemahan dan kelebihan yang tersendiri. Oleh itu, analisis yang teliti adalah perlu bagi memastikan metodologi pembangunan yang digunakan bersesuaian dengan pembangunan sistem.

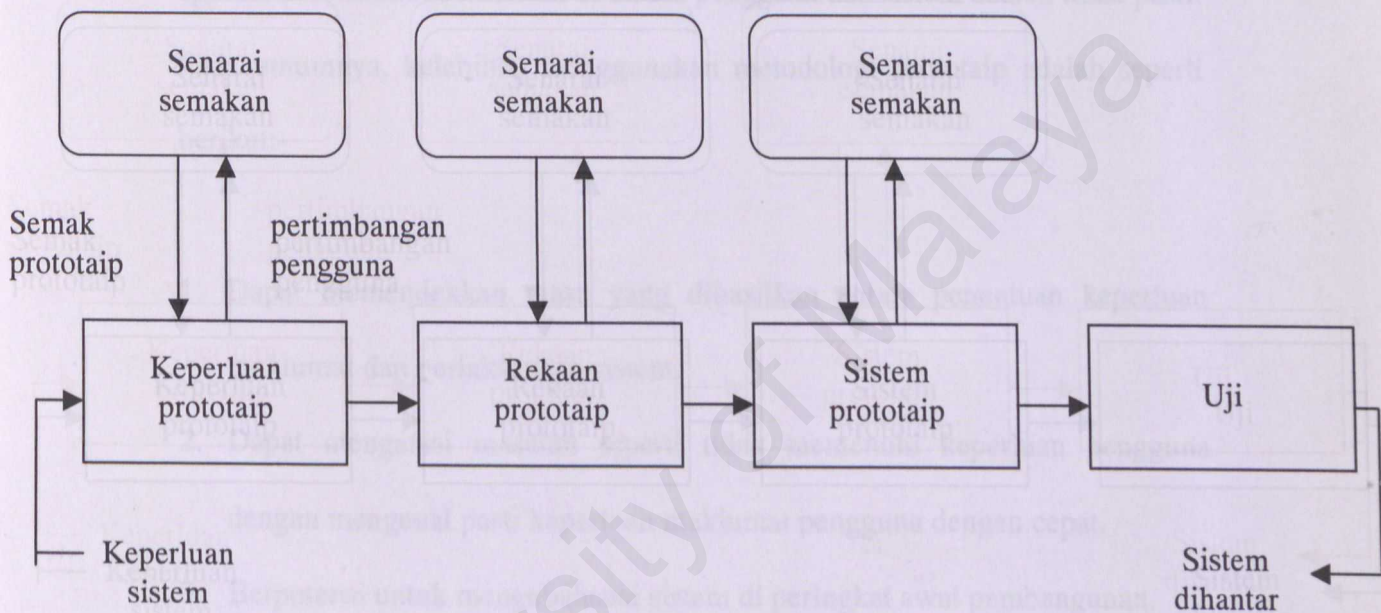
#### **3.2 METODOLOGI PEMBANGUNAN SISTEM**

Metodologi pembangunan memainkan peranan penting untuk memastikan sistem yang dibina adalah terkawal dan terurus. Tanpa metodologi yang sesuai, produk akhir yang terhasil tidak akan mencapai matlamat yang diinginkan. Di dalam pembangunan alat penterjemah Bahasa Melayu dan Bahasa Inggeris dan sebaliknya ini, model yang telah dipilih ialah model prototaip.



### 3.2.1 Model Prototaip

Model ini terdiri daripada 3 fasa yang utama iaitu fasa pengumpulan keperluan, reka bentuk perisian dan pembangunan prototaip. Ia sesuai digunakan apabila keperluan mendasar dan kelengkapan serta keperluan pemproses sistem masih memerlukan perubahan yang tidak pasti, pendekatan ini juga adalah lebih sesuai apabila kelengkapan interaksi di antara pengguna dan sistem adalah tidak pasti.



**Rajah 3.0 Model prototaip**

### 3.2.1 Model Prototaip

Model ini terdiri daripada 3 fasa yang utama iaitu fasa pengumpulan keperluan, rekabentuk pantas dan pembinaan prototaip. Ia sesuai digunakan apabila keperluan masukan dan keluaran serta keperluan pemproses sistem masih memerlukan perubahan yang tidak pasti, pendekatan ini juga adalah lebih sesuai apabila keberkesanan interaksi di antara pengguna dan sistem adalah tidak pasti. Pada umumnya, kelebihan menggunakan metodologi prototaip adalah seperti berikut:-

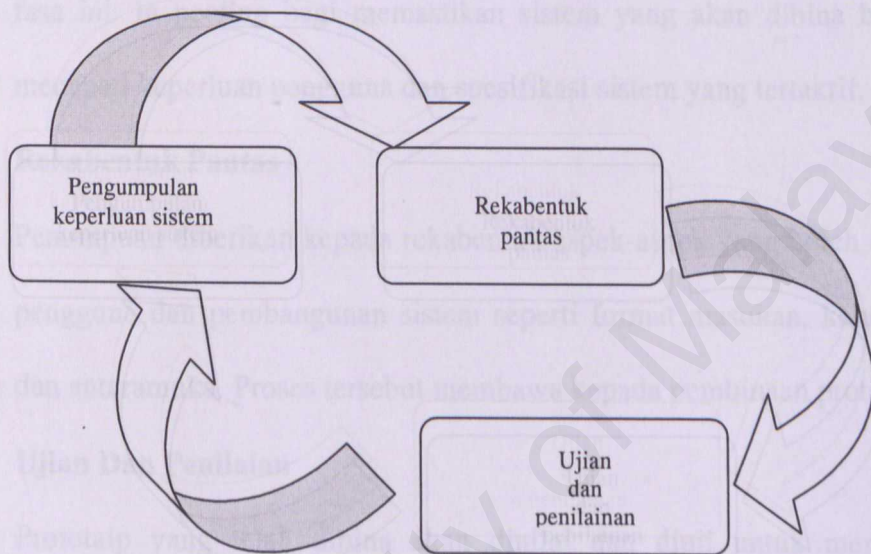
1. Dapat memendekkan masa yang dihasilkan antara penentuan keperluan maklumat dan perlaksanaan sistem.
2. Dapat mengatasi masalah seperti tidak memenuhi keperluan pengguna dengan mengenal pasti keperluan maklumat pengguna dengan cepat.
3. Berpotensi untuk mengubahsuai sistem di peringkat awal pembangunan.
4. Merekabentuk sistem yang memenuhi keperluan dan jangkaan pengguna
5. pengguna boleh melihat apa yang mungkin dan bagaimana keperluan yang dikehendaki dialihkan ke dalam bentuk perkakasan dan perisian.
6. mempunyai kesempatan untuk menghentikan pembangunan ke atas sistem yang tidak diperlukan.
7. Dapat mewujudkan kefahaman yang sama antara pembangun dan pengguna mengenai keperluan yang diinginkan.



Berikut adalah peneringan secara ringkas mengenai fasa-fasa dalam pembangunan model prototaip:-

### 1. Fasa Pengumpulan Keperluan Sistem

Keperluan sistem seperti objektif, skop dan keperluan data dikumpul pada fasa ini. Fasa ini memastikan sistem yang akan dibina benar-benar memenuhi keperluan pengguna dan spesifikasi sistem yang teraktir.



**Rajah 3.1 Fasa-fasa dalam model prototaip**

Berikut adalah penerangan secara ringkas mengenai fasa-fasa dalam pembangunan model prototaip:-

### **1. Fasa Pengumpulan Keperluan Sistem**

Keperluan sistem seperti objektif, skop dan keperluan data dikenal pasti pada fasa ini. Ia penting bagi memastikan sistem yang akan dibina benar-benar menepati keperluan pengguna dan spesifikasi sistem yang tertakrif.

### **2. Rekabentuk Pantas**

Penumpuan diberikan kepada rekabentuk aspek-aspek yang boleh dilihat oleh pengguna dan pembangunan sistem seperti format masukan, keluaran, skrin dan antaramuka. Proses tersebut membawa kepada pembinaan prototaip.

### **3. Ujian Dan Penilaian**

Prototaip yang telah dibina akan dinilai dan diuji untuk memastikan ia menepati spesifikasi yang telah ditakrifkan. Proses iteraktif atau pengulangan akan berlaku untuk memperhaluskan lagi prototaip yang telah dibina sehingga ke suatu tahap yang memuaskan pengguna dan pembangun sistem.

Prototaip menyediakan satu paradigma pembangunan yang berkesan kerana ia membenarkan proses pembaikan secara berulang dilakukan secara berterusan.



### 3.3 KAEDAH DAN TEKNIK

Teknik yang digunakan untuk menyelesaikan masalah ini adalah menggunakan teknik *Rule-Based System (RBS)*. *Rule-Based System (RBS)* menyimpan fakta-fakta yang diperlukan dalam suatu pangkalan data khas dikenali sebagai pangkalan perundangan (rule base). Biasanya, perundangan itu adalah dalam bentuk JIKA...MAKA. Sistem yang sedang dibangunkan ini memilih teknik *Rule-Based System (RBS)* sebagai pendekatan kerana ia bersesuaian dengan perisian yang dipilih iaitu *Visual Prolog* 5.2. Di mana *Visual Prolog* juga mempunyai ciri-ciri menggunakan sintaks JIKA...MAKA. Selain itu, *Rule-Based System (RBS)* boleh dikatakan menggunakan penyepadanan corak (pattern matching) untuk menyelesaikan masalah dengan bantuan rantaian ke hadapan dan rantaian ke belakang.

*Rule-Based System (RBS)* terdiri daripada 3 komponen utama seperti berikut:-

1. **Pangkalan perundangan (a rule base)**

Pangkalan perundangan ini terdiri daripada kesemua perundangan yang ada dalam sistem.

2. **Ingatan (working memory)**

Di mana ia berfungsi untuk menyimpan fakta-fakta permulaan dan keputusan atau hipotesis. Peranannya boleh disamakan seperti ingatan jangka pendek manusia.

3. **Pentafsir perundangan (a rule interpreter)**

Ia berperanan sebagai pengecam corak bagi menentukan perundangan mana yang perlu di 'fire' berdasarkan ingatan semasa.

algoritma bagi alat penterjemah ini adalah seperti berikut:-

1. Pengguna akan memasukkan teks atau dokumen yang akan diterjemahkan sama ada Bahasa Inggeris ke Bahasa Melayu dan sebaliknya. Sistem ini akan menentukan teks yang dimasukkan sama ada Bahasa Melayu atau Bahasa Inggeris.
2. Setelah teks diterima dalam bentuk rentetan (string), ia akan menentukan kategori bagi setiap perkataan daripada rentetan (string) dengan menyepadankan dengan pangkalan data dalaman sebelum ditukar kepada bentuk senarai (list).
3. Dua elemen senarai (list) pertama dibuat penyepadanan dengan pangkalan data dalaman terlebih dahulu, kemudian elemen seterusnya bagi menentukan makna, tatabahasa dan struktur.
4. Sistem akan menterjemahkan bahasa yang hendak diterjemahkan.



## BAB 4

### ANALISA DAN REKABENTUK SISTEM

#### 4.1 PENGENALAN

Sebagaimana yang kita ketahui, untuk membuat sesuatu perancangan projek beberapa persoalan yang harus dipertimbangkan bagi membangunkan sesebuah sistem. Keperluan sistem yang dimaksudkan meliputi keperluan dari segi keperluan fungsian, keperluan bukan fungsian dan keperluan teknikal yang terdiri daripada perkakasan dan perisian.

Menentukan keperluan secara tepat adalah sangat penting. Apabila keperluan-keperluan gagal ditakrifkan dengan tepat, implikasi yang akan timbul adalah seperti berikut:-

- Kos bagi pembangunan sistem akan melampaui kos asal yang dijangkakan
- Sistem tidak dapat disiapkan mengikut tempoh masa yang telah ditetapkan
- Mungkin akan menimbulkan ketidakpuashatian pengguna

Rekabentuk sistem merupakan satu proses di mana semua keperluan yang telah dikumpulkan dan dianalisa di dalam fasa pembangunan terdahulu digabungkan untuk membentuk model atau perwakilan suatu entiti iaitu alat penterjemah yang hendak digabungkan. Dalam erti kata yang lebih jelas, fasa rekabentuk sistem dilaksanakan untuk membangunkan suatu rekabentuk fizikal berasaskan rekabentuk logik sistem dengan menimbangkan keperluan-keperluan yang telah ditetapkan dalam fasa analisis sistem. Fasa ini menerangkan modul-modul yang terkandung di dalam sistem dan perhubungan yang wujud di antara modul dengan modul yang lain. Selain itu, cara

sistem dipersembahkan kepada pengguna juga ditunjukkan di dalam fasa ini. Rekabentuk antaramuka pengguna yang baik adalah kritikal bagi kejayaan sesuatu aplikasi. Pengguna umumnya menilai sesuatu sistem yang dibangunkan itu mengikut antaramukanya berbanding dengan kefungsiannya.

## **4.2 KEPERLUAN FUNGSIAN**

Menyatakan tentang fungsi-fungsi yang ditawarkan oleh sistem, bagaimana sistem bertindak balas terhadap sesuatu input dan juga cara kelakuan sistem dalam keadaan-keadaan tertentu.

Keperluan fungsian sistem ini adalah seperti berikut:-

### **1. Penerimaan dan pengecaman teks**

Pengguna akan memasukkan teks yang hendak diterjemahkan. Sistem akan mengkategorikan setiap perkataan daripada teks yang dimasukkan. Contoh kategori adalah seperti kata kerja (verb), kata nama (noun) dan lain-lain.

### **2. Pemeriksaan perkataan dalam pangkalan data**

Perkataan yang melibatkan sinonim, ambiguous dan lain-lain (Bahasa Melayu <-> Bahasa Inggeris) akan diperiksa dalam pangkalan data untuk mendapatkan penterjemahan yang tepat dari segi tatabahasa, makna dan struktur.



### 3. Menterjemah bahasa yang dipilih

Setelah melalui analisis serta pentafsiran makna dengan teliti bagi sesuatu ayat atau frasa, bahasa yang hendak diterjemahkan akan diterjemah berdasarkan program yang telah ditetapkan.

## 4.3 KEPERLUAN SPESIFIKASI

Keperluan spesifikasi yang telah ditentukan adalah seperti berikut:-

- Dapat menterjemah Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya
- Dapat menentukan tatabahasa yang betul, contohnya yang melibatkan *tense*
- Dapat menentukan kepunyaan (possession) dalam Bahasa Inggeris dengan betul
- Dapat menentukann *sense ambiguity* iaitu suatu perkataan yang berkemungkinan mempunyai lebih daripada satu makna.
- Dapat menentukan struktur ayat yang betul

## 4.4 KEPERLUAN BUKAN FUNGSIAN

Merujuk kepada ciri-ciri lain yang perlu ada pada sistem itu serta had-had atau halangan terhadap fungsi-fungsi yang ditawarkan oleh sistem. Ini termasuklah had-had yang wujud pada proses pembangunan sistem dan had masa. Keperluan bukan fungsian sistem ini adalah seperti berikut:-

### 1. Ketepatan

Merujuk kepada rekabentuk program yang memenuhi spesifikasi sistem dan keperluan pengguna

### 2. Kebolehpercayaan

Merujuk kepada sesuatu program itu boleh dijangkakan untuk mempamerkan fungsi yang dicadangkan

### **3. Kecekapan**

Merujuk kepada sesuatu prosedur boleh dipanggil dan dicapai berulang-ulang kali dan akan menghasilkan output yang sama

### **4. Kebolehgunaan**

Kebolehan program untuk mengendalikan, menyediakan input dan penafsiran output

### **5. Boleh Difahami**

Merujuk kepada untuk memahami aliran logik program tersebut. Dengan itu, perubahan boleh dibuat dengan mudah dalam bahagian aturcara yang perlu sahaja tanpa perlu mengubah logik yang lain

### **6. Boleh Selenggara**

Keupayaan untuk mengesan, mengubah dan menguji aplikasi program dikemaskini bagi memenuhi keperluan baru, melengkapkan kekurangan, membetulkan ralat atau berpindah ke sistem komputer yang lain

### **7. Saling Fungsi**

Berkeupayaan untuk berintegrasi dengan sistem atau program lain

### **8. Ramah Pengguna**

Ciri ramah pengguna merupakan aspek yang sangat penting dalam pembangunan sistem. Kejayaan menghasilkan sistem yang ramah pengguna akan menentukan kejayaan pembinaan sistem kerana secara realitinya pengguna tidak akan menggunakan sesuatu sistem yang sukar dioperasikan walaupun ia dibina dengan cara yang terbaik.



#### 4.5 KEPERLUAN TEKNIKAL

Untuk mengenalpasti perkakasan dan perisian yang manakah paling sesuai digunakan, ia bergantung kepada sejauh mana beban kerja yang akan dilakukan dalam rekabentuk dan mengimplementasikan sistem tersebut. Penggunaan perisian membantu pengaturcara dan pembangun sistem untuk membangunkan sebuah sistem dengan lebih baik, cekap, tepat, berkualiti dan dalam kos yang lebih murah dan munasabah.

Kita dapat menyaksikan perkembangan teknologi masa kini dengan kemunculan pelbagai perisian untuk pembangunan sistem terutamanya pengaturcaraan simbolik seperti LISP dan Visual Prolog. Untuk pembangunan suatu aplikasi adalah penting untuk seseorang pembangun aplikasi untuk melakukan beberapa pertimbangan sebelum memilih peralatan yang ingin digunakan. Antara perkara yang perlu diambil kira termasuklah:-

1. Perisian yang dipilih senang untuk dipelajari dan digunakan. Ini dapat memastikan masa untuk memahami dan mempelajari perisian tersebut adalah minimum dan aplikasi dapat disempurnakan dan dihantar pada masa yang ditetapkan.
2. Perisian tersebut mempunyai ciri-ciri yang diinginkan bagi sesebuah aplikasi yang ingin dibangunkan. Misalnya, jika aplikasi itu memerlukan capaian ke atas pangkalan data, maka perisian yang dipilih perlulah mempunyai keupayaan untuk membuat penyambungan ke pangkalan data. Dalam erti kata lain, perisian tersebut mampu berintegrasi dengan perisian yang lain.

3. Perisian yang dipilih perlulah berupaya untuk melakukan pelbagai jenis kerja pembangunan aplikasi termasuklah menjana kod-kod aturcara dan sesuai untuk pelbagai bahasa pengaturcaraan.
4. Dari sudut pemprototaipan kriteria untuk pemilihan perisian perlulah menjurus kepada kecekapan, kepantasan dan kemudahan untuk penggunaannya.

#### **4.5.1 Keperluan Perkakasan**

Perkakasan yang digunakan bagi membangunkan alat penterjemahan ini adalah komputer peribadi dengan spesifikasi minimum seperti berikut:-

Pemproses : Pentium 166 ke atas

Memori: 32 MB

Perkakasan input: papan kekunci, tetikus

Perkakasan output: monitor, pencetak

#### **4.5.2 Keperluan Perisian**

Perisian berikut telah dikenalpasti untuk digunakan sepanjang perlaksanaan rekabentuk dan pengimplementasian projek:-

- Visual Prolog 5.2

##### **4.5.2.1 Visual Prolog 5.2**

Prolog adalah pengaturcaraan yang menggunakan beberapa set mekanisma asas termasuklah, pattern matching, tree based data structuring and automatic back tracking. Prolog juga sesuai untuk



masalah yang melibatkan objek dan perhubungan. Perhubungan antara objek boleh dinyatakan dengan mudah menggunakan peraturan (Rule): *If object X is closer to the observer than object Y, and Y is closer than Z, then X must be closer than Z.* Visual Prolog sangat sesuai digunakan untuk perancangan Sistem Pakar atau masalah lain yang berkaitan dengan AI.

Visual Prolog dipilih sebagai bahasa pengaturcaraan utama dalam membangunkan projek ini disebabkan beberapa sebab berikut :-

- **Ciri-ciri sintaktiks dan sematiks**

Ciri-ciri prolog ini menggalakkan untuk memikirkan mengenai sesuatu masalah dan penyelesaiannya.

- **Pencarian Kebelakang (Backtracking Search)**

Prolog akan melakukan pencarian kebelakang secara automatik sekiranya perlu untuk mendapatkan keputusan atau hasil yang memuaskan.

- **Declarative meaning dan procedural meaning**

Declarative meaning dapat menentukan apakah output yang akan dihasilkan oleh suatu program itu. Procedural meaning pula menentukan bagaimana output itu boleh dihasilkan.

- **Masa Pembangunan Yang Singkat**

Disebabkan prolog adalah pengaturcaraan pengistiharan (declarative), bilangan baris dalam pengaturcaraan prolog untuk menyelesaikan masalah yang diberi adalah hanya sebahagian kecil yang diperlukan oleh bahasa pengaturcaraan prosedural. Ini akan mengurangkan kos pembangunan dan kod adalah mudah untuk diubahsuai. Kos penyelenggaraan juga didapati murah.

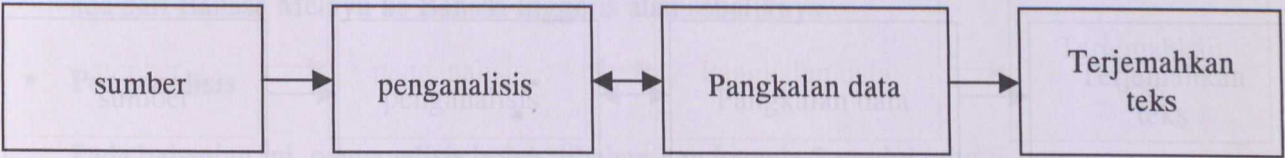
- **Mudah Memanipulasikan Struktur Data Yang Komplek**

Untuk pengaturcaraan prosedur, besar dan kompleks, aturcara sepatutnya boleh digunakan untuk menguruskan peruntukan ingatan apabila melaksanakan pengaturcaraan struktur data yang kompleks seperti pepohon, senarai atau graf dan penunjuk (pointer). Penunjuk (pointer) biasanya menyukarkan pengaturcaraan untuk dibaca. Prolog mempunyai notasi yang mudah dan elegen untuk pengulangan perterjemahan dan mengakses seperti struktur data dan pengurusan storan yang eksplisit.

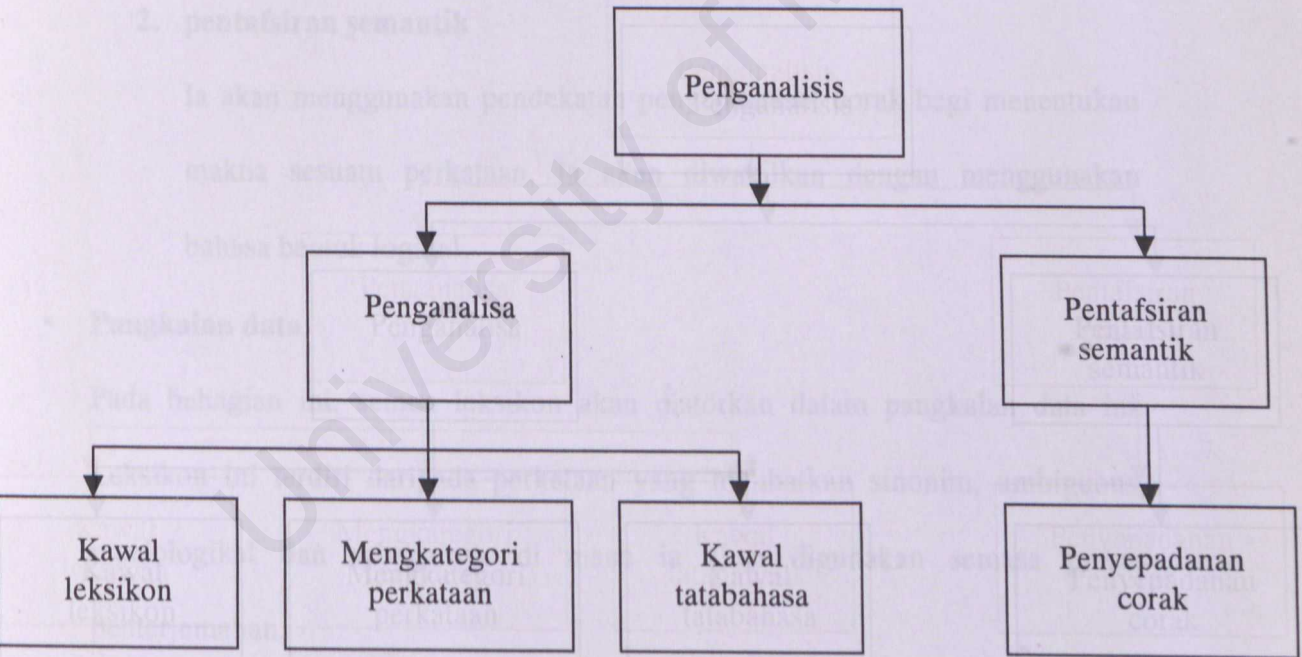


4.6 REKABENTUK STRUKTUR SISTEM

Pada rekabentuk struktur sistem ini, ia akan menunjukkan aliran maklumat dan modul-modul yang terlibat dalam pembangunan sistem penterjemah.



Rajah 4.0 Sturktur sistem penterjemah



Rajah 4.1 Modul-modul bagi penganalisis

Pada struktur sistem penterjemah di atas, terdapat 4 bahagian yang utama iaitu:-

- **Sumber**

Pengguna akan memasukkan teks dan bahasa yang hendak diterjemahkan sama ada dari Bahasa Melayu ke Bahasa Inggeris atau sebaliknya.

- **Penganalisis**

Pada bahagian ini, penganalisis boleh dibahagikan kepada 2 modul iaitu :-

1. **penganalisa**

Ia akan mempertimbangkan mengenai tatabahasa dan penghuraian di mana tatabahasa melibatkan penentuan objek jamak dan tunggal, analisis morfologikal dan leksikon, penentuan masa (tense) dan lain-lain.

2. **pentafsiran semantik**

Ia akan menggunakan pendekatan penyepadanan corak bagi menentukan makna sesuatu perkataan. Ia akan diwakilkan dengan menggunakan bahasa bentuk logikal.

- **Pangkalan data**

Pada bahagian ini, semua leksikon akan distorkan dalam pangkalan data ini. Leksikon ini terdiri daripada perkataan yang melibatkan sinonim, ambiguous, morfologikal dan sebagainya di mana ia akan digunakan semasa proses penterjemahan.

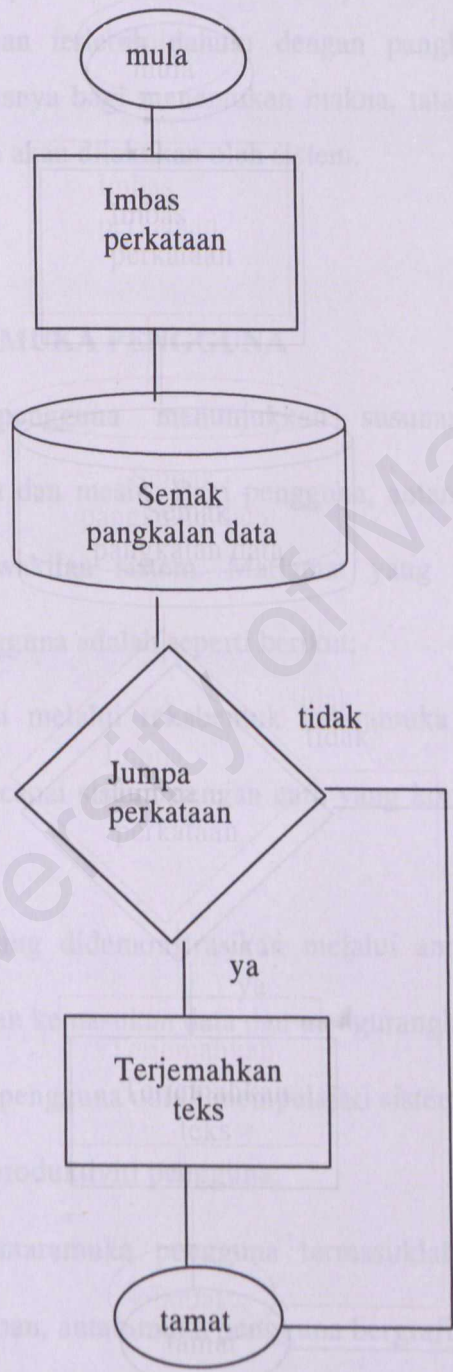
- **Terjemahkan teks**

Pada bahagian ini, ia akan menterjemahkan teks berdasarkan bahasa yang hendak diterjemahkan



4.7 ALIRAN MAKLUMAT

Aliran maklumat mengenai sistem penterjemah ditunjukkan seperti berikut:-



Rajah 4.2 Aliran maklumat sistem penterjemah

Pengguna akan memasukkan teks atau dokumen yang hendak diterjemahkan. Setelah teks diterima dalam bentuk rentetan (string), ia akan menentukan kategori bagi setiap perkataan daripada rentetan (string) dengan menyepadankan dengan pangkalan data dalaman sebelum ditukar kepada bentuk senarai (list). Dua elemen senarai (list) pertama dibuat penyepadanan terlebih dahulu dengan pangkalan data dalaman kemudian dua elemen seterusnya bagi menentukan makna, tatabahasa dan struktur. Kemudian terjemahan bahasa akan dilakukan oleh sistem.

#### 4.8 REKABENTUK ANTARAMUKA PENGGUNA

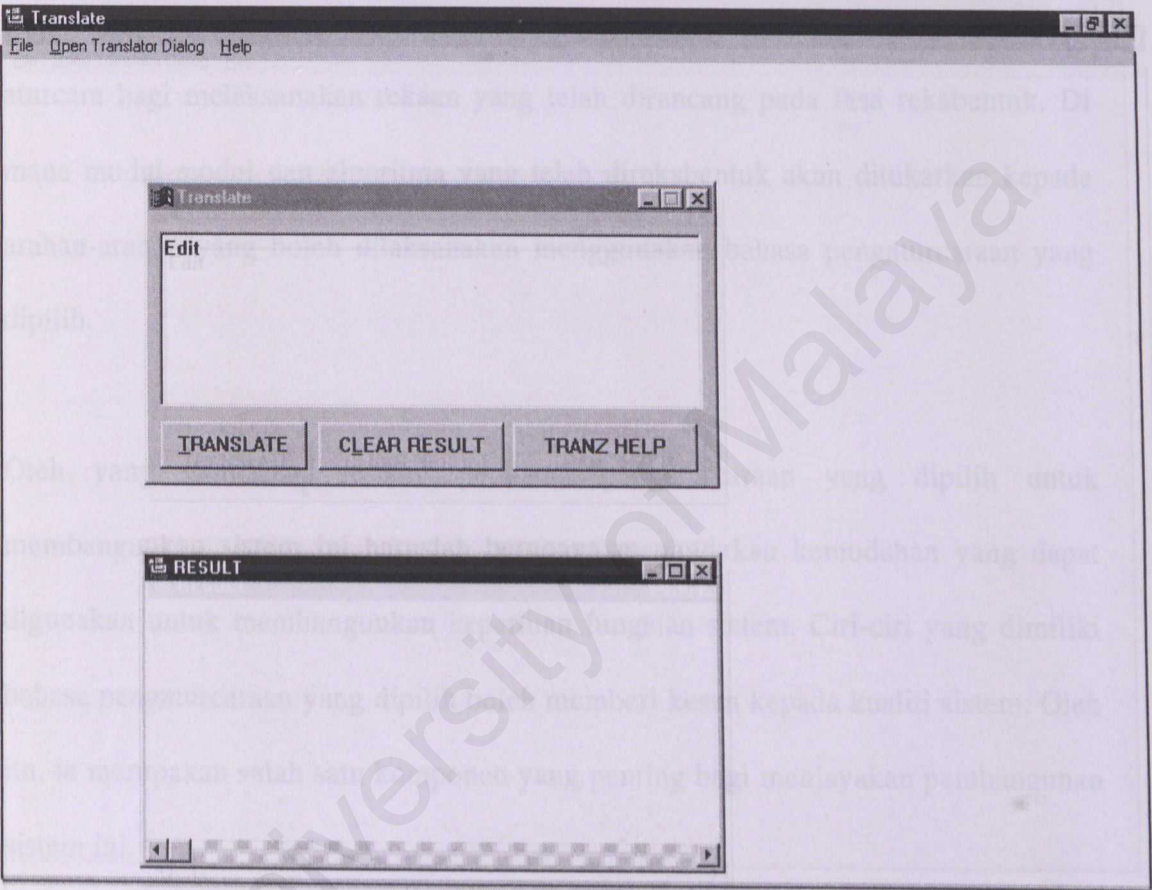
Rekabentuk antaramuka pengguna menunjukkan susunan dan mekanisme perhubungan antara manusia dan mesin. Bagi pengguna, antaramuka adalah sistem dan dianggap sebagai perwakilan sistem. Matlamat yang perlu dicapai untuk membentuk antaramuka pengguna adalah seperti berikut:-

1. keberkesanan dicapai melalui rekabentuk antaramuka yang membenarkan pengguna untuk mencapai sistem dengan cara yang kongruen sejajar dengan keperluan individu.
2. kecekapan seperti yang didemonstrasikan melalui antaramuka yang boleh meningkatkan kelajuan kemasukan data dan mengurangkan ralat.
3. mengurangkan masa pengguna untuk mempelajari sistem tersebut.
4. meningkatkan katar produktiviti pengguna.

Terdapat beberapa jenis antaramuka pengguna termasuklah antaramuka bahasa semulajadi, soalan dan jawapan, antaramuka pengguna bergrafik (GUI) dan lain-lain. Kebanyakan sistem pada hari ini menggunakan antaramuka pengguna bergrafik (GUI). Ianya mudah dipelajari dan digunakan walaupun bagi pengguna yang kurang



biasa dengan sistem komputer. Elemen-elemen yang tersedia pada GUI seperti tetingkap, menu, pull-down menu, scroll bar, imej ikon dan banyak lagi dapat membuatkan antaramuka pengguna itu lebih bersifat ramah pengguna.



Rajah 4.3 Antaramuka pengguna bagi sistem penterjemah

## **BAB 5**

### **PERLAKSANAAN/PEMBANGUNAN SISTEM**

#### **5.1 PENGENALAN**

Selepas fasa rekabentuk sistem selesai, proses pelaksanaan atau pembangunan sistem akan dijalankan. Pelaksanaan atau pembangunan sistem ini melibatkan penulisan aturcara bagi melaksanakan rekaan yang telah dirancang pada fasa rekabentuk. Di mana modul-modul dan algoritma yang telah direkabentuk akan ditukarkan kepada arahan-arahan yang boleh dilaksanakan menggunakan bahasa pengaturcaraan yang dipilih.

Oleh yang demikian, sesuatu peralatan pengaturcaraan yang dipilih untuk membangunkan sistem ini haruslah berupaya menawarkan kemudahan yang dapat digunakan untuk membangunkan keperluan fungsian sistem. Ciri-ciri yang dimiliki bahasa pengaturcaraan yang dipilih boleh memberi kesan kepada kualiti sistem. Oleh itu, ia merupakan salah satu komponen yang penting bagi menjayakan pembangunan sistem ini.

#### **5.2 PENGKODAN**

Pengkodan merupakan penukaran modul-modul atau algoritma ke dalam arahan-arahan (aturcara) yang membolehkan keperluan fungsian sistem yang dicadangkan dapat dilaksanakan.



Pengkodan bagi alat penterjemahan ini adalah seperti berikut:-

### **Bahagian 1**

Pada bahagian 1 ini, penerimaan dan pengecaman teks daripada dialog translate akan dilakukan kemudian menghantarnya kepada bahagian seterusnya.

diff(\_Win):-

```
consult("translate.db",db),
```

```
A=win_getctlhandle(_Win, IDC_EDIT),
```

```
STR=win_gettext(A),
```

```
tokl(STR,TOKL),
```

```
s_sentence(TOKL).
```

### **Bahagian 2**

Pada bahagian 2 ini, setelah teks diterima dalam bentuk rentetan (string), ia akan menentukan kategori bagi setiap perkataan daripada rentetan (string) dengan menyepadankan dengan pangkalan data dalaman sebelum ditukar kepada bentuk senarai (list).

tokl(STR,[TOK|TOKL]) :-

```
fronttoken(STR,TOK,STR1),
```

```
check(TOK),!,
```

```
tokl(STR1,TOKL).
```

```
tokl(_,[]).
```

check(X):-

```
noun(X,_),!.
```

### Bahagian 3

Pada bahagian 3 ini menunjukkan senarai (list) yang terhasil.

```
s_sentence(TOKL):-  
    s_split(TOKL).
```

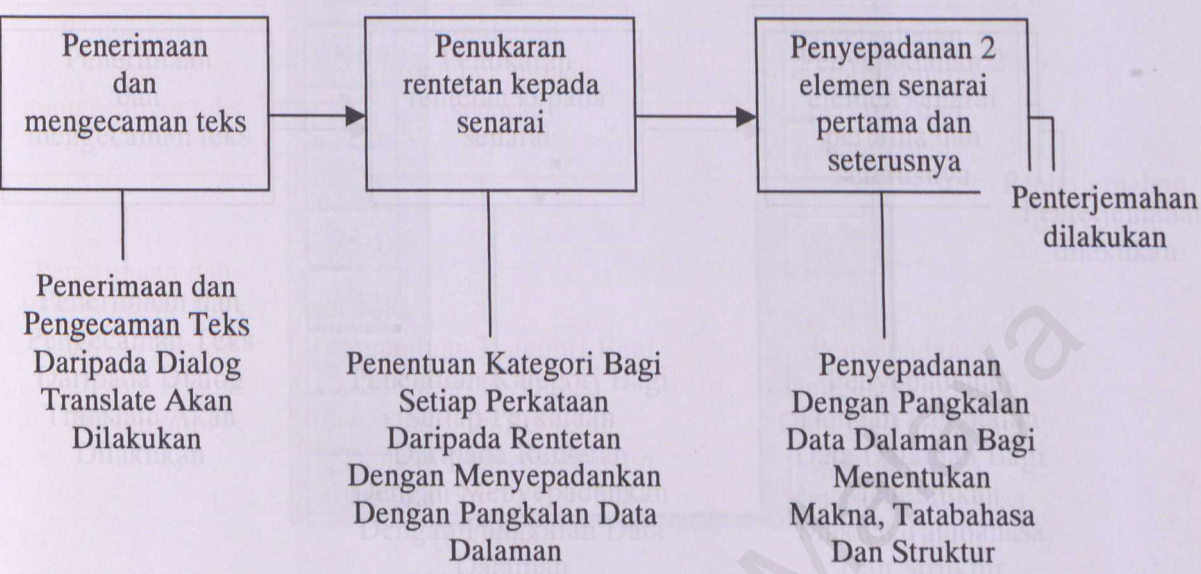
### Bahagian 4

Pada bahagian 4 ini, dua elemen senarai (list) pertama dibuat penyepadanan dengan pangkalan data dalaman bagi menentukan makna, tatabahasa dan struktur.

```
s_split([REL,REL2|TOKL]):-  
    check1(REL,REL2),  
    s_split([REL2|TOKL]).
```

```
check1(REL,REL2):-  
    pron(X,REL),verb3(_ ,REL2),!,write(X," ").  
  
check1(REL,REL2):-  
    pron(X,REL),verb(Y,REL2),!,write(X," "),write(Y," ").
```

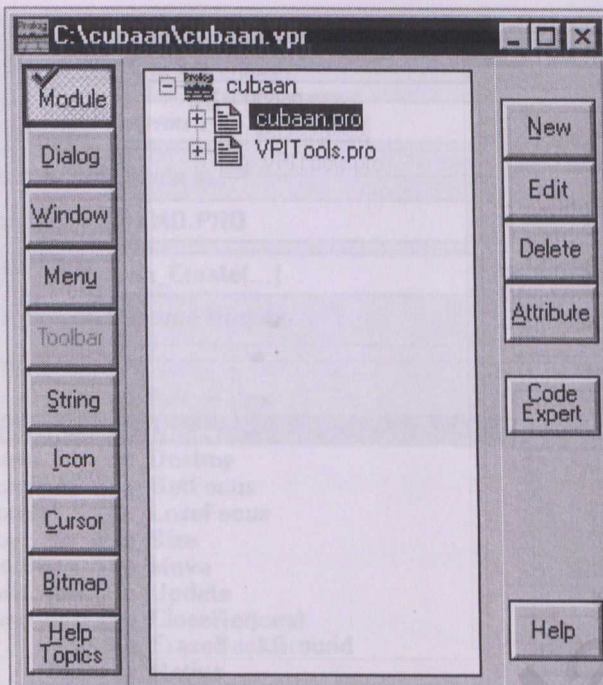




**Rajah 5.0 Menunjukkan perlaksanaan sistem dengan lebih terperinci**

**5.3 PERSEKITARAN PEMBANGUNAN**

Bahasa pengaturcaraan yang dipilih bagi membangunkan sistem ini adalah *Visual Prolog 5.2*. Perisian *Visual Prolog 5.2* dibekalkan dengan beberapa komponen seperti *Project Window* dan *The Dialog and Window Expert* bagi memudahkan pengaturcara semasa menulis aturcara.

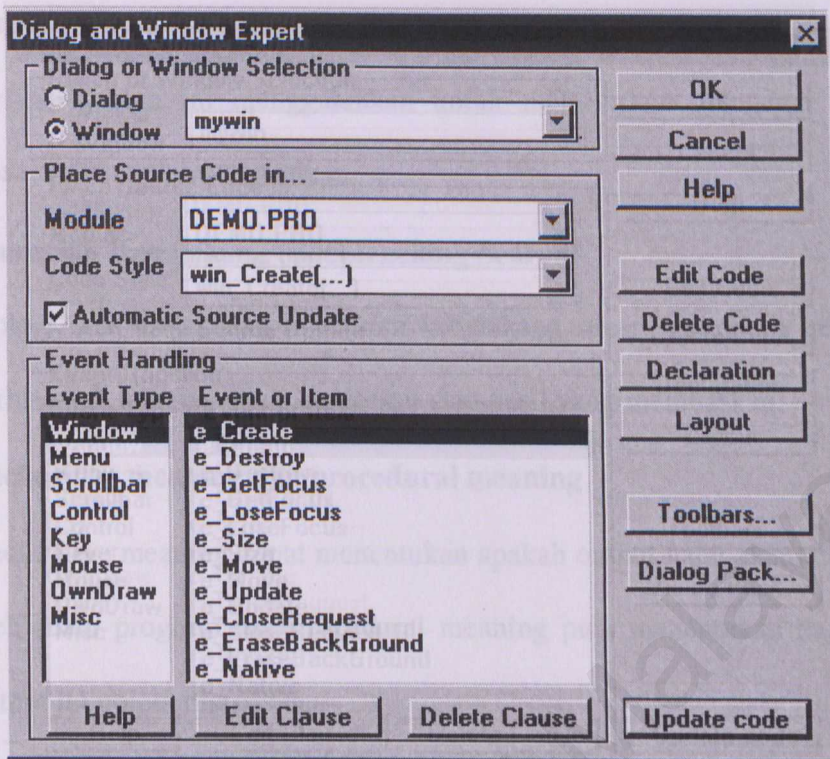


**Rajah 5.1 Project Window**

### 5.3.1 Project Window

*Project Window* merupakan capaian utama kepada komponen aplikasi Visual Prolog. Komponen-komponen yang terdapat dalam *Project Window* adalah seperti modul, dialog, window, menu dan toolbar. Kesemua kod-kod aturcara akan ditulis pada bahagian modul bagi membolehkan kefungsi sistem dapat dilaksanakan.





**Rajah 5.2 The Dialog and Window Expert**

### 5.3.2 The Dialog and Window Expert

*The Dialog and Window Expert* merupakan peralatan yang menghubungkan kod Prolog kepada dialog dan window yang telah direka. *The Dialog and Window Expert* akan memasukkan kod Prolog yang diperlukan untuk mengawal dialog dan window yang telah direka itu. Ini akan menjimatkan masa dan tenaga pengaturcara kerana pengkodan bagi dialog dan window telah diautomasikan pada peralatan *The Dialog and Window Expert*.

Visual Prolog dipilih sebagai bahasa pengaturcaraan utama dalam membangunkan projek ini disebabkan beberapa sebab berikut :-

- **Ciri-ciri sintaktiks dan semantiks**

Ciri-ciri prolog ini menggalakkan untuk memikirkan mengenai sesuatu masalah dan penyelesaiannya.

- **Pencarian Kebelakang (Backtracking Search)**

Prolog akan melakukan pencarian kebelakang secara automatik sekiranya perlu untuk mendapatkan keputusan atau hasil yang memuaskan.

- **Declarative meaning dan procedural meaning**

Declarative meaning dapat menentukan apakah output yang akan dihasilkan oleh suatu program itu. Procedural meaning pula menentukan bagaimana output itu boleh dihasilkan.

- **Masa Pembangunan Yang Singkat**

Disebabkan prolog adalah pengaturcaraan pengistiharan (declarative), bilangan baris dalam pengaturcaraan prolog untuk menyelesaikan masalah yang diberi adalah hanya sebahagian kecil yang diperlukan oleh bahasa pengaturcaraan prosedural. Ini akan mengurangkan kos pembangunan dan kod adalah mudah untuk diubahsuai. Kos penyelenggaraan juga didapati murah.

- **Mudah Memanipulasikan Struktur Data Yang Komplek**

Untuk pengaturcaraan prosedur, besar dan komplek, aturcara sepatutnya boleh digunakan untuk menguruskan peruntukan ingatan apabila melaksanakan pengaturcaraan struktur data yang komplek seperti pepohon, senarai atau graf dan penunjuk (pointer). Penunjuk (pointer) biasanya



menyukarkan pengaturcaraan untuk dibaca. Prolog mempunyai notasi yang mudah dan elegan untuk pengulangan perterjemahan dan mengakses seperti struktur data dan pengurusan storan yang eksplisit.

Contoh pengkodan bagi dialog yang telah diautomasikan:-

```
%BEGIN_DLG translate
/*****
Creation and event handling for dialog: translate
*****/

constants

%BEGIN translate, CreateParms, 15:47:18-8.1.2003, Code automatically updated!

dlg_translate_ResID = idd_translate
dlg_translate_DlgType = wd_Modal
dlg_translate_Help = idh_contents

%END translate, CreateParms

predicates

dlg_translate_eh : EHANDLER
dlg_translate_handle_answer(INTEGER EndButton,DIALOG_VAL_LIST)
dlg_translate_update(DIALOG_VAL_LIST)
```

clauses

dlg\_translate\_Create(Parent):-

%MARK translate, new variables

dialog\_CreateModal(Parent,dlg\_translate\_ResID,"",

[

%BEGIN translate, ControlList, 15:47:18-8.1.2003, Code automatically updated!

df(idc\_edit,editstr("Edit",[]),nopr)

%END translate, ControlList

],

dlg\_translate\_eh,0,VALLIST,ANSWER),

dlg\_translate\_handle\_answer(ANSWER,VALLIST).

dlg\_translate\_handle\_answer(idc\_ok,VALLIST):-!,

dlg\_translate\_update(VALLIST).

dlg\_translate\_handle\_answer(idc\_cancel,\_):-!. % Handle Esc and Cancel here

dlg\_translate\_handle\_answer(\_):-

errorexit().

dlg\_translate\_update(\_VALLIST):-

%BEGIN translate, Update controls, 15:47:18-8.1.2003, Code automatically updated!

\_IDC\_EDIT\_VALUE = dialog\_VLGetstr(idc\_edit,\_VALLIST),



%END translate, Update controls

true.

%MARK translate, new events

%BEGIN translate, idc\_translate \_CtlInfo

dlg\_translate\_eh(\_Win,e\_Control(idc\_translate,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-!,

diff(\_Win),!.

%END translate, idc\_translate \_CtlInfo

%BEGIN translate, idc\_padam \_CtlInfo

dlg\_translate\_eh(\_Win,e\_Control(idc\_padam,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-!,

msg\_clear(),!.

%END translate, idc\_padam \_CtlInfo

dlg\_translate\_eh(\_,\_,\_):-!,fail.

%END\_DLG translate

## BAB 6

### PENGUJIAN SISTEM

#### 6.1 PENGENALAN

Pada bahagian pengujian sistem ini bertujuan untuk mengesan ralat yang menyebabkan suatu sistem itu mengalami kegagalan. Selain itu, ia juga berfungsi untuk mengesahkan sama ada sistem berfungsi mengikut keperluan dan spesifikasi yang telah ditetapkan. Pengenalpastian ralat akan dilakukan bagi menentukan apakah ralat yang menyebabkan kegagalan. Sekiranya ralat dapat dikesan, pembetulan ralat akan dijalankan di mana ia melibatkan proses membuat perubahan kepada sistem untuk membuang ralat tersebut.

Keperluan spesifikasi yang telah ditentukan adalah seperti berikut:-

- Dapat menterjemah Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya
- Dapat menentukan tatabahasa yang betul, contohnya yang melibatkan *tense*
- Dapat menentukan kepunyaan (possession) dalam Bahasa Inggeris dengan betul
- Dapat menentukann *sense ambiguity* iaitu suatu perkataan yang berkemungkinan mempunyai lebih daripada satu makna.
- Dapat menentukan struktur ayat yang betul

#### 6.2 STRATEGI PENGUJIAN

Bagi mengenalpasti ralat yang wujud dan mengesahkan sama ada sistem berfungsi mengikut keperluan dan spesifikasi yang telah ditentukan, terdapat empat strategi



telah digunakan bagi menjalankan pengujian sistem ini. Empat strategi tersebut adalah seperti berikut:-

- Pengujian modul
- Pengujian integrasi
- Pengujian fungsi
- Pengujian prestasi
- Pengujian sistem

## 6.2.2 Pengujian Integrasi

### 6.2.1 Pengujian Modul

Pengujian modul atau unit berfungsi untuk mengesahkan setiap komponen atau berfungsi dengan betul. Pada sistem penterjemah ini terdapat tiga modul adalah seperti berikut:-

#### Modul 1

Untuk menerima dan mengecam teks yang hendak diterjemahkan. Ia akan diuji bagi memastikan bahawa ia dapat mengecam teks yang hendak diterjemahkan.

#### Modul 2

Ia akan menentukan kategori bagi setiap perkataan daripada rentetan (string) dengan menyepadankan dengan pangkalan data dalaman sebelum ditukar kepada bentuk senarai (list). Ia akan diuji sama ada setiap perkataan daripada rentetan itu dapat dikenalpasti kategorinya atau tidak.

### **Modul 3**

Daripada senarai (list), dua elemen senarai (list) pertama dibuat penyepadanan dengan pangkalan data dalaman. Modul ini merupakan modul yang terpenting kerana di sinilah makna, tatabahasa dan struktur ditentukan. Ia akan diuji sama ada sistem ini dapat menentukan makna, tatabahasa dan struktur dengan betul.

#### **6.2.2 Pengujian Integrasi**

Pengujian integrasi merupakan proses yang mengesahkan ketiga-tiga modul sistem bekerjasama sebagaimana yang telah dinyatakan di dalam sistem dan juga spesifikasi rekabentuk sistem. Apabila maklumat yang dihantar pada komponen mengikut rekabentuk, ia akan uji sistem untuk memastikan ia mempunyai fungsi yang diinginkan.

#### **6.2.3 Pengujian Fungsi**

Pengujian fungsi akan menilai sistem untuk menentukan sama ada fungsi yang telah dinyatakan dalam spesifikasi keperluan telah dilaksanakan di dalam sistem yang telah diintegrasikan tadi.

#### **6.2.4 Pengujian Prestasi**

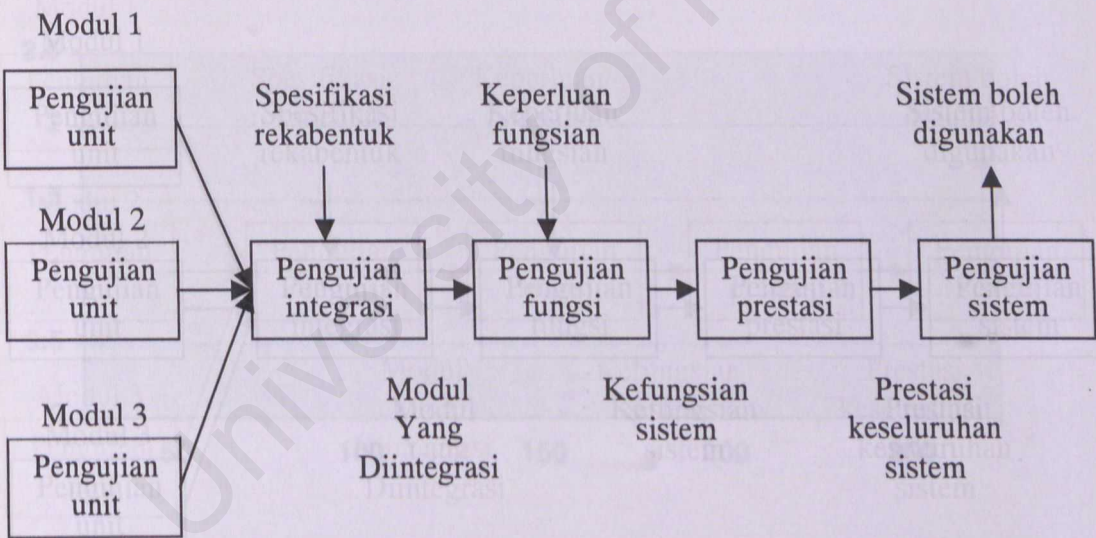
Ia membandingkan komponen yang telah diintegrasikan dengan keperluan bukan fungsian sistem. Ini lebih merujuk kepada kamus di mana semua perkataan dalam Bahasa Melayu dan Bahasa Inggeris yang disimpan di



dalam pangkalan data dalaman. Ia akan menguji adakah simpanan kamus yang banyak akan melambatkan masa untuk sistem ini membuat penterjemahan. Selain itu, ia juga akan menguji adakah bilangan teks yang banyak akan membebankan sistem ini untuk melakukan penterjemahan.

6.2.5 Pengujian sistem

Objektif bagi pengujian sistem adalah bagi memastikan sistem melaksanakan apa yang telah dinyatakan dalam spesifikasi. Pengujian sistem menunjukkan prestasi keseluruhan sistem ini.



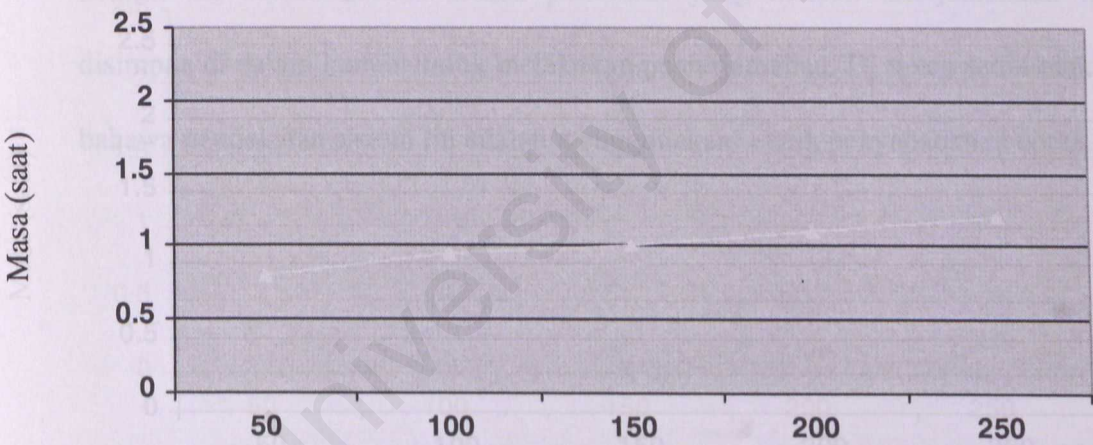
Rajah 6.0 Langkah-langkah pengujian yang dijalankan

6.3 ANALISA KEPUTUSAN UJIAN

Saiz kamus (unit)	0	50	100	150	200	250
Masa (Saat)	0	0.8	0.95	1	1.1	1.2

Jadual 1.0 Menunjukkan kebergantungan antara masa dan saiz

kamus yang diperlukan untuk membuat penterjemahan



Saiz kamus (1 unit penterjemahan)

Rajah 6.1 Menunjukkan kebergantungan antara masa dan saiz Mamus yang diperlukan untuk membuat penterjemahan



Pada sistem ini, masa tindak balas dengan saiz kamus adalah saling bergantung. Pada Jadual dan Rajah di atas dapat kita lihat bahawa sekiranya 50 perkataan dalam kamus memerlukan masa 0.8 saat untuk membuat penterjemahan. Seterusnya 100 perkataan dalam kamus memerlukan masa 0.95 saat, 150 perkataan dalam kamus memerlukan masa 1 saat, 200 perkataan dalam kamus memerlukan masa 1.1 saat dan 250 perkataan dalam kamus memerlukan masa 1.2 saat. Berdasarkan data-data yang diperoleh, dapatlah disimpulkan bahawa semakin banyak bilangan perkataan yang disimpan di dalam kamus, semakin banyak masa yang diperlukan untuk membuat penterjemahan. Selain itu, bilangan perkataan dalam teks yang hendak diterjemahkan juga berkadar langsung dengan bilangan saiz perkataan dalam kamus. Ini adalah kerana semua perkataan yang hendak diterjemahkan telah disimpan di dalam kamus untuk melakukan penterjemahan. Di mana sedia maklum bahawa pendekatan sistem ini adalah menggunakan teknik penyepadanan corak.

## BAB 7

### PERBINCANGAN

#### 7.1 KEPUTUSAN DAN KESIMPULAN YANG DIPEROLEH

Sistem ini merupakan alat penterjemah Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya. Ia dibangunkan menggunakan perisian *Visual Prolog 5.2*. Sistem ini dibangunkan dengan menggunakan pendekatan teknik *Rule-Based System* di mana ia lebih kepada penyepadanan corak dengan bantuan rantai ke hadapan dan rantai ke belakang. Sistem ini hanya dapat menterjemah teks yang mudah sahaja. Oleh itu, ia tidak sesuai untuk penterjemahan teks yang kompleks. Pada sistem ini, masa tindak balas dengan saiz kamus adalah saling bergantung. Selain itu, saiz kamus dengan bilangan teks yang hendak diterjemahkan juga saling bergantung. Sistem ini tidak melibatkan penghuraian pepohon seperti yang digunakan oleh sesetengah sistem penterjemah yang sedia ada. Pada sistem ini, ia akan menentukan kategori bagi setiap perkataan daripada rentetan (string) dengan menyepadankan dengan pangkalan data dalaman sebelum ditukar kepada bentuk senarai (list). Dua elemen senarai (list) pertama dibuat penyepadanan dengan pangkalan data dalaman terlebih dahulu, kemudian dua elemen seterusnya bagi menentukan makna, tatabahasa dan struktur. Tujuan ia dilakukan sedemikian adalah untuk menentukan perkaitan antara dua perkataan tersebut. Selepas itu, sistem akan menterjemahkan bahasa yang hendak diterjemahkan.



## 7.2 KELEBIHAN SISTEM YANG DIBANGUNKAN

Sistem yang dibangun ini mempunyai kelebihan seperti berikut:-

- Dapat menterjemah Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya
- Dapat menentukan tatabahasa yang betul, contohnya yang melibatkan *tense*
- Dapat menentukan kepunyaan (possession) dalam Bahasa Inggeris dengan betul
- Dapat menentukann *sense ambiguity* iaitu suatu perkataan yang berkemungkinan mempunyai lebih daripada satu makna.
- Dapat menentukan struktur ayat yang betul

## 7.3 MASALAH/KELEMAHAN SISTEM YANG DIBANGUNKAN

Sistem ini mempunyai kelebihan yang tersendiri. Di samping itu, ia juga mempunyai kelemahan seperti berikut:-

- Sistem ini tidak melibatkan penghuraian pepohon seperti yang digunakan oleh sesetengah sistem yang sedia ada. Sistem ini hanya melibatkan penukaran teks dalam bentuk rentetan kepada bentuk senarai. Kemudian, dua elemen senarai pertama dibuat penyepadanan dengan pangkalan data dalaman terlebih dahulu, kemudian dua elemen seterusnya bagi menentukan makna, tatabahasa dan struktur.
- Sistem ini memang dapat menentukan tatabahasa dengan betul, contohnya melibatkan *tense*. Tetapi, sistem ini tidak dapat menentukan tatabahasa dengan betul sepenuhnya, hanya sesetengah sahaja.

- Sistem ini memang dapat menentukann *sense ambiguity* iaitu suatu perkataan yang berkemungkinan mempunyai lebih daripda satu makna. Tetapi, sistem ini tidak dapat melakukan sedemikian sepenuhnya.
- Pada sistem ini, mungkin terdapat perkataan yang tidak dapat diterjemahkan. Ini mungkin disebabkan perkataan tersebut tidak terdapat dalam kamus pada pangkalan data dalaman. Ini disebabkan kerja memasukkan data ke dalam pangkalan data dalaman memerlukan masa yang panjang.

#### 7.4 PENINGKATAN YANG DIJALANKAN TERHADAP SISTEM

Masalah tatabahasa yang melibatkan *tense* dan masalah *sense ambiguity* boleh diatasi dengan membaiki bentuk leksikon atau kamus serta isi aturcara. Ini dapat dilakukan dengan hanya menambah fungsi penentu ke dalam kamus dan aturcara. Contohnya bagi masalah *sense ambiguity* bagi perkataan “saw”, ia boleh memberi maksud lihat atau menggergaji. Bagi mengatasi masalah ini, perkataan “saw” dipecahkan kepada beberapa fungsi penentu sebagai verb1 dan verb2. Cara mengatsi masalah yang melibatkan tatabahasa sama seperti sebagaimana masalah *sense ambiguity* diatasi. Masalah perkataan yang tidak dapat diterjemahkan oleh sistem boleh diatasi dengan menambah bilangan perkataan yang mungkin ke dalam kamus.



## APENDIKS A (PENGKODAN)

Copyright (c) DERIS Company

Project: TRANSLATE

FileName: TRANSLATE.PRO

Purpose: Translate Language BM <-> BI

Written by: Mohd Idris Hj Odding

```
*****  
include "translate.inc"  
include "translate.con"
```

### DATABASE-db

```
adj( STRING,STRING )  
det( STRING,STRING )  
det1( STRING,STRING )  
det2( STRING,STRING )  
noun( STRING,STRING )  
noun1( STRING,STRING )  
verb( STRING,STRING )  
verb1( STRING,STRING )  
verb2( STRING,STRING )  
verb3( STRING,STRING )  
verb4( STRING,STRING )  
verb5( STRING,STRING )  
pron( STRING,STRING )  
pron1( STRING,STRING )  
pron2( STRING,STRING )  
pron3( STRING,STRING )  
prep( STRING,STRING )  
punct( STRING )  
noun3( STRING )
```

### DOMAINS

```
TOKL = STRING*
```

### PREDICATES

```
again(STRING)  
check(STRING)  
tokl( STRING, TOKL )  
nondeterm diff(window)  
nondeterm check1(string,string)
```

```

nondeterm s_split( TOKL)
nondeterm s_sentence( TOKL)
CLAUSES
diff(_ Win):-
    consult("translate.db",db),          %PANGKALAN DATA DALAMAN DI MANA
PERKATAAN-PERKATAAN DISIMPAN
    A=win_getctlhandle(_ Win,idc_edit),  %MENERIMA DAN MENGECEK TEKS
YANG ADA PADA DIALOG TRANSLATE
    STR=win_gettext(A),
    tokl(STR,TOKL),                      %MENUMUKARKAN RENTETAN (STRING)
KEPADA SENARAI (LIST)
    s_sentence(TOKL).                    %SENARAI (LIST) YANG TERHASIL

%MENUMUKARKAN RENTETAN (STRING) KEPADA SENARAI (LIST)
tokl(STR,[TOK|TOKL]) :-
    fronttoken(STR,TOK,STR1),
    check(TOK),!,
    tokl(STR1,TOKL).
tokl(_,[]).

/*CHECK - MENENTUKAN KATEGORI BAGI SETIAP PERKATAAN DARIPADA
RENTETAN (STRING) DENGAN MENYEPADANKAN
DENGAN PANGKALAN DATA DALAMAN*/
check(X):-
    noun3(X),!.

check(X):-
    noun(X,_),!.
check(X):-
    noun(_,X),!.

check(X):-
    noun1(_,X),!.

check(X):-
    noun1(X,_),!.

check(X):-
    verb(X,_),!.

check(X):-
    verb(_,X),!.

check(X):-

```



verb1(X,\_),!.

check(X):-  
verb1(\_X),!.

check(X):-  
verb2(X,\_),!.

check(X):-  
verb2(\_X),!.

check(X):-  
verb3(X,\_),!.

check(X):-  
verb3(\_X),!.

check(X):-  
verb4(X,\_),!.

check(X):-  
verb4(\_X),!.

check(X):-  
verb5(X,\_),!.

check(X):-  
verb5(\_X),!.

check(X):-  
det(\_X),!.

check(X):-  
det(X,\_),!.

check(X):-  
det1(\_X),!.

check(X):-  
det1(X,\_),!.

check(X):-  
det2(\_X),!.

check(X):-  
det2(X,\_),!.

```

check(X):-
    prep(_X),!.

check(X):-
    prep(X,_),!.

check(X):-
    pron(_X),!.

check(X):-
    pron(X,_),!.

check(X):-
    pron1(_X),!.

check(X):-
    pron1(X,_),!.

check(X):-
    pron2(_X),!.

check(X):-
    pron2(X,_),!.

check(X):-
    pron3(_X),!.

check(X):-
    pron3(X,_),!.

check(X):-
    punct(X),!.

check(X):-
    adj(_X),!.

check(X):-
    adj(X,_),!.

check(X):-
    write(">> Unknown word: ",X),nl.

```



%SENARAI (LIST) YANG TERHASIL

s\_sentence(TOKL):-

s\_split(TOKL).

/\*DUA ELEMEN SENARAI (LIST) PERTAMA DAN SETERUSNYA DIBUAT  
PENYEPADANAN DENGAN PANGKALAN DATA DALAMAN  
BAGI MENENTUKAN MAKNA, TATABAHASA DAN STRUKTUR\*/

s\_split([REL,REL2|TOKL]):-

check1(REL,REL2),

s\_split([REL2|TOKL]).

%BI->BM

check1(REL,REL2):-

punct(REL),noun3(REL2),!.

check1(REL,REL2):-

punct(REL),noun(\_,REL2),!.

check1(REL,REL2):-

punct(REL),pron(\_,REL2),!.

check1(REL,REL2):-

punct(REL),prep(\_,REL2),!.

check1(REL,REL2):-

punct(REL),verb2(\_,REL2),!.

check1(REL,REL2):-

punct(REL),pron1(\_,REL2),!.

check1(REL,REL2):-

pron(X,REL),verb3(\_,REL2),!,write(X," ").

check1(REL,REL2):-

pron(X,REL),verb(Y,REL2),!,write(X," "),write(Y," ").

check1(REL,REL2):-

pron(X,REL),verb5(Y,REL2),!,write(X," "),write(Y," ").

check1(REL,REL2):-

pron(X,REL),verb4(Y,REL2),!,write(X," "),write(Y," ").

check1(REL,REL2):-  
     pron(X,REL),verb4(Y,REL2),!,write(X," "),write(Y," ").

check1(REL,REL2):-  
     pron1(X,REL),verb3(\_ ,REL2),!,write(X," ").

check1(REL,REL2):-  
     pron1(X,REL),verb(Y,REL2),!,write(X," "),write(Y," ").

check1(REL,REL2):-  
     pron1(X,REL),verb4(Y,REL2),!,write(X," "),write(Y," ").

check1(REL,REL2):-  
     pron2(\_ ,REL),!,again(REL2).

check1(REL,REL2):-  
     pron2(\_ ,REL),prep(\_ ,REL2),!.

check1(REL,REL2):-  
     pron2(X,REL2),verb(\_ ,REL),!,write(X," ").

check1(REL,REL2):-  
     pron2(X,REL),noun(Y,REL2),!,write(Y," "),write(X," ").

check1(REL,REL2):-  
     pron3(X,REL),noun(Y,REL2),!,write(Y," "),write(X," ").

check1(REL,REL2):-  
     pron3(X,REL),verb2(Y,REL2),!,write(Y," "),write(X," ").

check1(REL,REL2):-  
     pron3(X,REL),verb1(Y,REL2),!,write(Y," "),write(X," ").

check1(REL,REL2):-  
     verb(\_ ,REL),verb(Y,REL2),!,write(Y," ").

check1(REL,REL2):-  
     verb(\_ ,REL),verb1(\_ ,REL2),!.

check1(REL,REL2):-  
     verb(\_ ,REL),!,again(REL2).

check1(REL,REL2):-  
     verb1(\_ ,REL),!,again(REL2).



check1(REL,REL2):-  
     X="adakah",  
     verb1(X,REL),pron(\_ ,REL2),!,write(X," ").

check1(REL,REL2):-  
     X="adakah",  
     verb1(X,REL),noun3(REL2),!,write(X," ").

check1(REL,REL2):-  
     verb1(\_ ,REL),verb(Y,REL2),!,write(Y," ").

check1(REL,REL2):-  
     verb1(\_ ,REL),verb5(Y,REL2),!,write(Y," ").

check1(REL,REL2):-  
     verb2(X,REL),pron1(\_ ,REL2),!,write(X," ").

check1(REL,REL2):-  
     verb3(\_ ,REL),verb5(Y,REL2),!,write(Y," ").

check1(REL,REL2):-  
     verb3(\_ ,REL),verb(Y,REL2),!,write(Y," ").

check1(REL,REL2):-  
     verb4(\_ ,REL),!,again(REL2).

check1(REL,REL2):-  
     verb5(\_ ,REL),!,again(REL2).

check1(REL,REL2):-  
     verb5(\_ ,REL),punct(REL2),!,write(REL2," "),nl.

check1(REL,REL2):-  
     prep(X,REL),noun(Y,REL2),!,write(X," "),write(Y," ").

check1(REL,REL2):-  
     prep(X,REL),pron(\_ ,REL2),!,write(X," ").

check1(REL,REL2):-  
     prep(X,REL),pron1(\_ ,REL2),!,write(X," ").

check1(REL,REL2):-  
     prep(X,REL),det(\_ ,REL2),!,write(X," ").

check1(REL,REL2):-  
     noun(X,REL),noun3(REL2),!,write(X," ").  
 check1(REL,REL2):-  
     noun(X,REL),pron(\_ ,REL2),!,write(X," ").  
 check1(REL,REL2):-  
     noun(X,REL),verb(Y,REL2),!,write(X," "),write(Y," ").  
 check1(REL,REL2):-  
     noun(\_ ,REL),verb4(Y,REL2),!,write(Y," ").  
 check1(REL,REL2):-  
     noun(X,REL),pron1(\_ ,REL2),!,write(X," ").  
 check1(REL,REL2):-  
     noun(\_ ,REL),adj(\_ ,REL2),!.  
 check1(REL,REL2):-  
     noun(\_ ,REL),punct(REL2),!,write(REL2," "),nl.  
 check1(REL,REL2):-  
     noun3(REL),punct(REL2),!,write(REL2," "),nl.  
 check1(REL,REL2):-  
     noun3(REL),verb(Y,REL2),!,write(REL," "),write(Y," ").  
 check1(REL,REL2):-  
     noun3(REL),verb4(Y,REL2),!,write(REL," "),write(Y," ").  
 check1(REL,REL2):-  
     noun3(REL),prep(\_ ,REL2),!.  
 check1(REL,REL2):-  
     noun3(REL),verb3(\_ ,REL2),!,write(REL," ").  
 check1(REL,REL2):-  
     adj(X,REL),noun(Y,REL2),!,write(Y," "),write(X," ").  
 check1(REL,REL2):-  
     det(\_ ,REL),noun(\_ ,REL2),!.  
 check1(REL,REL2):-  
     det(X,REL),verb(Y,REL2),!,write(X," "),write(Y," ").



%BM->BI

check1(REL,REL2):-  
    punct(REL),verb1(REL2,\_),!.

check1(REL,REL2):-  
    punct(REL),verb2(REL2,\_),!.

check1(REL,REL2):-  
    punct(REL),pron(REL2,\_),!.

check1(REL,REL2):-  
    punct(REL),pron1(REL2,\_),!.

check1(REL,REL2):-  
    punct(REL),prep(REL2,\_),!.

check1(REL,REL2):-  
    punct(REL),noun3(REL2),!.

check1(REL,REL2):-  
    REL2="akan",  
    pron(REL,X),!,write(X," "),verb(REL2,Y),!,write(Y," ").

check1(REL,REL2):-  
    pron(REL,X),verb5(REL2,Y),!,write(X," ","was "),write(Y," ").

check1(REL,REL2):-  
    pron(REL,X),verb4(REL2,\_),!,write(X," ").

check1(REL,REL2):-  
    pron(REL,X),det(REL2,\_),!,write(X," ").

check1(REL,REL2):-  
    pron(REL,X),verb(REL2,Y),!,write(X," "),concat(Y,"s",Z),write(Z," ").

check1(REL,REL2):-  
    pron1(REL,X),verb5(REL2,Y),!,write(X," ","were "),write(Y," ").

check1(REL,REL2):-  
    pron1(REL,X),det(REL2,\_),!,write(X," ").

check1(REL,REL2):-  
    pron1(REL,\_),noun(REL2,\_),!.

check1(REL,REL2):-  
     pron1(REL,X),verb(REL2,Y),!,write(X," "),write(Y," ").  
 check1(REL,REL2):-  
     pron1(REL,X),verb4(REL2,\_),!,write(X," ").  
  
 check1(REL,REL2):-  
     pron3(REL,\_),verb4(REL2,Y),!,write(Y," ").  
  
 check1(REL,REL2):-  
     pron3(REL,\_),noun(REL2,\_),!.  
  
 check1(REL,REL2):-  
     pron3(REL,\_),punct(REL2),!,write(REL2," "),nl.  
  
 check1(REL,REL2):-  
     verb(REL,\_),verb(REL2,Y),!,write(Y," ").  
  
 check1(REL,REL2):-  
     verb(REL,\_),pron2(REL2,Y),!,write(Y," ").  
  
 check1(REL,REL2):-  
     verb(REL,\_),noun3(REL2),!,write(REL2," ").  
  
 check1(REL,REL2):-  
     verb(REL,\_),punct(REL2),!,write(REL2," "),nl.  
  
 check1(REL,REL2):-  
     verb(REL,\_),prep(REL2,\_),!.  
  
 check1(REL,REL2):-  
     verb(REL,\_),noun(REL2,Y),!,write(Y," ").  
  
 check1(REL,REL2):-  
     verb(REL,\_),verb5(REL2,Y),!,write("be "),write(Y," ").  
  
 check1(REL,REL2):-  
     X="has",  
     verb1(REL,X),pron(REL2,\_),!,write(X," ").  
  
 check1(REL,REL2):-  
     X="has",  
     verb1(REL,X),noun3(REL2),!,write(X," ").  
  
 check1(REL,REL2):-  
     verb2(REL,X),pron1(REL2,\_),!,write(X," ").



check1(REL,REL2):-  
     verb4(REL,X),noun(REL2,Y),!,write(X," "),write(Y," ").  
 check1(REL,REL2):-  
     verb4(REL,\_),punct(REL2),!,write(REL2),nl.  
  
 check1(REL,REL2):-  
     verb5(REL,\_),noun(REL2,\_),!.  
  
 check1(REL,REL2):-  
     verb5(REL,\_),punct(REL2),!,write(REL2),nl.  
  
 check1(REL,REL2):-  
     adj(REL,\_),!,again(REL2).  
  
 check1(REL,REL2):-  
     prep(REL,X),noun(REL2,Y),!,write(X," "),write(Y," ").  
  
 check1(REL,REL2):-  
     prep(REL,X),pron1(REL2,\_),!,write(X," ").  
  
 check1(REL,REL2):-  
     prep(REL,X),noun3(REL2),!,write(X," ").  
  
 check1(REL,REL2):-  
     prep(REL,X),pron(REL2,\_),!,write(X," ").  
  
 check1(REL,REL2):-  
     noun(REL,\_),verb(REL2,Y),!,write(Y," ").  
  
 check1(REL,REL2):-  
     noun(REL,X),adj(REL2,Y),!,write(Y," "),write(X," ").  
  
 check1(REL,REL2):-  
     noun(REL,\_),punct(REL2),!,write(REL2," "),nl.  
  
 check1(REL,REL2):-  
     noun(REL,X),pron3(REL2,Y),!,write(Y," "),write(X," ").  
  
 check1(REL,REL2):-  
     noun(REL,X),noun3(REL2),!,write(X," ").  
  
 check1(REL,REL2):-  
     REL2="akan",  
     noun3(REL),!,write(REL," "),verb(REL2,Y),!,write(Y," ").  
  
 check1(REL,REL2):-

noun3(REL),verb(REL2,Y),!,write(REL," "),concat(Y,"s",Z),write(Z," ").

check1(REL,REL2):-

noun3(REL),det(REL2,\_),!,write(REL," ").

check1(REL,REL2):-

noun3(REL),verb4(REL2,\_),!,write(REL," ").

check1(REL,REL2):-

noun3(REL),verb5(REL2,Y),!,write(REL," ","was "),write(Y," ").

check1(REL,REL2):-

noun3(REL),prep(REL2,\_),!.

check1(REL,REL2):-

det(REL,X),verb(REL2,Y),!,write(Y," "),write(X," ").

%BI->BM

again(REL2):-

noun(Y,REL2),!,write(Y," ").

again(REL2):-

prep(\_ ,REL2),!.

again(REL2):-

pron(\_ ,REL2),!.

again(REL2):-

pron1(\_ ,REL2),!.

again(REL2):-

pron2(Y,REL2),!,write(Y," ").

again(REL2):-

pron2(\_ ,REL2),!.

again(REL2):-

pron3(\_ ,REL2),!.

again(REL2):-

det(\_ ,REL2),!.

again(REL2):-

punct(REL2),!,write(REL2," "),nl.



%BEGIN\_WIN Task Window

/\*\*\*\*\*\*

Event handling for Task Window

\*\*\*\*\*/

predicates

task\_win\_eh : EHANDLER

constants

%BEGIN Task Window, CreateParms, 19:59:56-17.2.2003, Code automatically updated!

task\_win\_Flags =

[wsf\_SizeBorder,wsf\_TitleBar,wsf\_Close,wsf\_Maximize,wsf\_Minimize,wsf\_ClipSibling  
s,wsf\_Maximized,wsf\_TopMost]

task\_win\_Menu = res\_menu(idr\_task\_menu)

task\_win\_Title = "Translate"

task\_win\_Help = contents

%END Task Window, CreateParms

clauses

%BEGIN Task Window, e\_Create

task\_win\_eh(\_Win,e\_Create(\_),0):-!,

%BEGIN Task Window, InitControls, 19:59:56-17.2.2003, Code automatically updated!

%END Task Window, InitControls

%BEGIN Task Window, ToolbarCreate, 19:59:56-17.2.2003, Code automatically  
updated!

%END Task Window, ToolbarCreate

ifdef use\_message

msg\_Create(100,"RESULT"),

endif

!.

%END Task Window, e\_Create

%MARK Task Window, new events

%BEGIN Task Window, id\_open\_translator\_dialog

task\_win\_eh(\_Win,e\_Menu(id\_open\_translator\_dialog, ShiftCtlAlt),0):-!,

dlg\_translate\_Create(\_Win),dlg\_attention\_Create(\_Win),!.

%END Task Window, id\_open\_translator\_dialog

%BEGIN Task Window, id\_help\_contents

```

task_win_eh(_Win,e_Menu(id_help_contents,_ShiftCtlAlt),0):-!,
    vpi_ShowHelp("translate.hlp"),
    !.
%END Task Window, id_help_contents

%BEGIN Task Window, id_help_about
task_win_eh(Win,e_Menu(id_help_about,_ShiftCtlAlt),0):-!,
    dlg_about_dialog_Create(Win),
    !.
%END Task Window, id_help_about

%BEGIN Task Window, id_file_exit
task_win_eh(Win,e_Menu(id_file_exit,_ShiftCtlAlt),0):-!,
    win_Destroy(Win),
    !.
%END Task Window, id_file_exit

%BEGIN Task Window, e_Size
task_win_eh(_Win,e_Size(_Width,_Height),0):-!,
#ifdef use_tbar
    toolbar_Resize(_Win),
#endif
#ifdef use_message
    msg_Resize(_Win),
#endif
    !.
%END Task Window, e_Size

%END_WIN Task Window

/*****
Invoking on-line Help
*****/

project_ShowHelpContext(HelpTopic):-
    vpi_ShowHelpContext("translate.hlp",HelpTopic).

/*****
Main Goal
*****/

goal

#ifdef use_mdi
    vpi_SetAttrVal(attr_win_mdi,b_true),
#endif

```



```

ifndef ws_win
  ifdef use_3dctrl
    vpi_SetAttrVal(attr_win_3dcontrols,b_true),
  endif
endif
vpi_Init(task_win_Flags,task_win_eh,task_win_Menu,"translate",task_win_Title).

%BEGIN_DLG About dialog
/*****
Creation and event handling for dialog: About dialog
*****/

constants

%BEGIN About dialog, CreateParms, 22:49:57-4.2.2003, Code automatically updated!
  dlg_about_dialog_ResID = idd_dlg_about
  dlg_about_dialog_DlgType = wd_Modal
  dlg_about_dialog_Help = contents
%END About dialog, CreateParms

predicates

  dlg_about_dialog_eh : EHANDLER

clauses

  dlg_about_dialog_Create(Parent):-
    win_CreateResDialog(Parent,dlg_about_dialog_DlgType,dlg_about_dialog_ResID,
    dlg_about_dialog_eh,0).

%BEGIN About dialog, idc_ok_CtlInfo
  dlg_about_dialog_eh(_Win,e_Control(idc_ok,_CtrlType,_CtrlWin,_CtrlInfo),0):-!,
    win_Destroy(_Win),
    !.
%END About dialog, idc_ok_CtlInfo
%MARK About dialog, new events

  dlg_about_dialog_eh(_,_):-!,fail.

%END_DLG About dialog

```

%BEGIN\_DLG Translate

/\*\*\*\*\*\*

Creation and event handling for dialog: Translate

\*\*\*\*\*/

constants

%BEGIN Translate, CreateParms, 20:38:32-17.2.2003, Code automatically updated!

dlg\_translate\_ResID = idd\_translate

dlg\_translate\_DlgType = wd\_Modeless

dlg\_translate\_Help = contents

%END Translate, CreateParms

predicates

dlg\_translate\_eh : EHANDLER

clauses

dlg\_translate\_Create(Parent):-

win\_CreateResDialog(Parent,dlg\_translate\_DlgType,dlg\_translate\_ResID,dlg\_translate\_eh,0).

%MARK Translate, new events

%BEGIN Translate, idc\_tranzhelp\_CtlInfo

dlg\_translate\_eh(\_Win,e\_Control(idc\_tranzhelp,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-!,  
dlg\_tranz\_help\_Create(\_Win),!.

%END Translate, idc\_tranzhelp\_CtlInfo

%BEGIN Translate, idc\_translate\_CtlInfo

dlg\_translate\_eh(\_Win,e\_Control(idc\_translate,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-!,  
diff(\_Win),!.

%END Translate, idc\_translate\_CtlInfo

%BEGIN Translate, idc\_clear\_messages\_CtlInfo

dlg\_translate\_eh(\_Win,e\_Control(idc\_clear\_messages,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-!,  
msg\_clear(),!.

%END Translate, idc\_clear\_messages\_CtlInfo



```

    dlg_translate_eh(,_,_):-!,fail.
%END_DLG Translate
%BEGIN_DLG TRANZ Help
/*****
    Creation and event handling for dialog: TRANZ Help
*****/

```

## constants

```

%BEGIN TRANZ Help, CreateParms, 13:22:57-17.2.2003, Code automatically updated!
    dlg_tranz_help_ResID = idd_tranz_help
    dlg_tranz_help_DlgType = wd_Modal
    dlg_tranz_help_Help = contents
%END TRANZ Help, CreateParms

```

## predicates

```

dlg_tranz_help_eh : EHANDLER
dlg_tranz_help_handle_answer(INTEGER EndButton,DIALOG_VAL_LIST)
dlg_tranz_help_update(DIALOG_VAL_LIST)

```

## clauses

```

dlg_tranz_help_Create(Parent):-

```

```

%MARK TRANZ Help, new variables

    dialog_CreateModal(Parent,dlg_tranz_help_ResID,"",
    [
%BEGIN TRANZ Help, ControlList, 13:22:57-17.2.2003, Code automatically updated!
        df(idc_untuk_menggunakan_sistem_ini_pengguna,editstr("Untuk
Menggunakan Sistem ini, Pengguna",[[]],nopr),
        df(idc_tranz_help_3,editstr("Perlu Menggunakan Huruf Kecil
Untuk",[[]],nopr),
        df(idc_tranz_help_4,editstr("Kesemua Perkataan Kecuali Huruf
Permulaan",[[]],nopr),
        df(idc_tranz_help_5,editstr("Perkataan Bagi Kata Nama Khas",[[]],nopr),
        df(idc_tranz_help_7,editstr("-->klik Butang Ini Untuk Menterjemahkan
Teks",[[]],nopr),
        df(idc_tranz_help_8,editstr(" Yang Dimasukkan",[[]],nopr),
        df(idc_tranz_help_10,editstr("-->klik Butang Ini Untuk Memadamkan
Hasil",[[]],nopr),
        df(idc_tranz_help_11,editstr(" Penterjemahan",[[]],nopr)
%END TRANZ Help, ControlList
    ],
    dlg_tranz_help_eh,0,VALLIST,ANSWER),

```

```
dlg_tranz_help_handle_answer(ANSWER,VALLIST).
```

```
dlg_tranz_help_handle_answer(idc_ok,VALLIST):-!,
```

```
    dlg_tranz_help_update(VALLIST).
```

```
dlg_tranz_help_handle_answer(idc_cancel,_):-!. % Handle Esc and Cancel here
```

```
dlg_tranz_help_handle_answer(_):-
```

```
    errexit().
```

```
dlg_tranz_help_update(_VALLIST):-
```

```
%BEGIN TRANZ Help, Update controls, 13:22:57-17.2.2003, Code automatically  
updated!
```

```
    _IDC_UNTUK_MENGGUNAKAN_SISTEM_INI_PENGGUNA_VALUE =  
    dialog_VLGetstr(idc_untuk_menggunakan_sistem_ini_pengguna,_VALLIST),
```

```
    _IDC_TRANZ_HELP_3_VALUE =
```

```
    dialog_VLGetstr(idc_tranz_help_3,_VALLIST),
```

```
    _IDC_TRANZ_HELP_4_VALUE =
```

```
    dialog_VLGetstr(idc_tranz_help_4,_VALLIST),
```

```
    _IDC_TRANZ_HELP_5_VALUE =
```

```
    dialog_VLGetstr(idc_tranz_help_5,_VALLIST),
```

```
    _IDC_TRANZ_HELP_7_VALUE =
```

```
    dialog_VLGetstr(idc_tranz_help_7,_VALLIST),
```

```
    _IDC_TRANZ_HELP_8_VALUE =
```

```
    dialog_VLGetstr(idc_tranz_help_8,_VALLIST),
```

```
    _IDC_TRANZ_HELP_10_VALUE =
```

```
    dialog_VLGetstr(idc_tranz_help_10,_VALLIST),
```

```
    _IDC_TRANZ_HELP_11_VALUE =
```

```
    dialog_VLGetstr(idc_tranz_help_11,_VALLIST),
```

```
%END TRANZ Help, Update controls
```

```
    true.
```

```
%MARK TRANZ Help, new events
```

```
dlg_tranz_help_eh(_,_):-!,fail.
```

```
%END_DLG TRANZ Help
```

```
%BEGIN_DLG Attention
```

```
/******
```

```
    Creation and event handling for dialog: Attention
```

```
*****/
```

```
constants
```

```
%BEGIN Attention, CreateParms, 21:12:57-17.2.2003, Code automatically updated!
```

```
dlg_attention_ResID = idd_attention
```

```
dlg_attention_DlgType = wd_Modal
```



```
dlg_attention_Help = contents
%END Attention, CreateParms
predicates
```

```
dlg_attention_eh : EHANDLER
dlg_attention_handle_answer(INTEGER EndButton,DIALOG_VAL_LIST)
dlg_attention_update(DIALOG_VAL_LIST)
```

clauses

```
dlg_attention_Create(Parent):-
```

```
%MARK Attention, new variables
```

```
    dialog_CreateModal(Parent,dlg_attention_ResID,"",
    [
%BEGIN Attention, ControlList, 21:12:57-17.2.2003, Code automatically updated!
%END Attention, ControlList
    ],
    dlg_attention_eh,0,VALLIST,ANSWER),
    dlg_attention_handle_answer(ANSWER,VALLIST).
```

```
dlg_attention_handle_answer(idc_ok,VALLIST):-!,
    dlg_attention_update(VALLIST).
dlg_attention_handle_answer(idc_cancel,_):-!. % Handle Esc and Cancel here
dlg_attention_handle_answer(_):-
    errexit().
```

```
dlg_attention_update(_VALLIST):-
%BEGIN Attention, Update controls, 21:12:57-17.2.2003, Code automatically updated!
%END Attention, Update controls
    true.
```

```
%MARK Attention, new events
```

```
dlg_attention_eh(_,_):-!,fail.
```

```
%END_DLG Attention
```

## APENDIKS B (PANKALAN DATA DALAMAN)

det(" ", "the")  
det("banyak", "many")  
det("sepanjang", "all")  
det("dan", "and")  
det("atau", "or")  
det("setiap", "every")  
det("dan", "and")  
det("atau", "or")  
det("tidak", "not")  
det("sebatang", "a")  
det1("seorang", "a")  
det2("seekor", "a")  
pron("ia", "it")  
pron("dia", "he")  
pron("dia", "she")  
pron1("kamu", "you")  
pron1("saya", "i")  
pron1("kami", "we")  
pron1("mereka", "they")  
pron2("mereka", "them")  
pron2("saya", "me")  
pron2("kami", "us")  
pron2("kamu", "you")  
pron2("dia", "him")  
pron2("ia", "it")  
pron3("ini", "this")  
pron3("itu", "that")  
pron3("siapa", "who")  
pron3("apa", "what")  
pron3("mana", "where")  
pron3("bagaimana", "how")  
pron3("saya", "my")  
pron3("kamu", "your")  
pron3("ia", "its")  
pron3("mereka", "their")  
pron3("kami", "our")  
pron3("dia", "his")  
noun("rumah", "home")  
noun("sementara", "while")  
noun("isteri", "wife")  
noun("suami", "husband")  
noun("berakhir", "ended")  
noun("kakak", "sister")  
noun("abang", "brother")



noun("datuk", "grandfatar")  
 noun("nenek", "grandmother")  
 noun("cucu", "grandchild")  
 noun("bapa", "father")  
 noun("ibu", "mother")  
 noun("gadis", "girl")  
 noun("bunga", "flower")  
 noun("bakat", "talent")  
 noun("ayam", "chicken")  
 noun("pagi", "morning")  
 noun("petang", "evening")  
 noun("malam", "night")  
 noun("pokok", "tree")  
 noun("kepala", "head")  
 noun("hidung", "nose")  
 noun("rambut", "hair")  
 noun("sungai", "river")  
 noun("ancaman", "threat")  
 noun("lelaki", "man")  
 noun("budak", "boy")  
 noun("anak", "child")  
 noun("aduan", "complaint")  
 noun("agama", "religion")  
 noun("islam", "islamic")  
 noun("kristian", "christiant")  
 noun("peperiksaan", "exam")  
 noun("air", "water")  
 noun("ahli", "member")  
 noun("gajah", "elephant")  
 noun("ayam", "chicken")  
 noun("duit", "money")  
 noun("buaya", "crocodile")  
 noun("belalai", "trunk")  
 noun("kenderaan", "transport")  
 noun("basikal", "bicycle")  
 noun("motosikal", "motorcycle")  
 noun("kereta", "car")  
 noun("lori", "lorry")  
 noun("badan", "body")  
 noun("tangan", "hand")  
 noun("kaki", "feet")  
 noun("kepala", "head")  
 noun("telinga", "ears")  
 noun("mulut", "mouth")  
 noun("hidung", "nose")  
 noun("mata", "eyes")

noun("anugerah", "award")  
 noun("arahan", "instruction")  
 noun("arah", "direction")  
 noun("angin", "wind")  
 noun("awan", "cloud")  
 noun("matahari", "sun")  
 noun("api", "fire")  
 noun("bulan", "moon")  
 noun("bintang", "star")  
 noun("bukit", "hill")  
 noun("gunung", "mountain")  
 noun("almari", "cupboard")  
 noun("alkohol", "alcohol")  
 noun("televisyen", "television")  
 noun("bantal", "pillow")  
 noun("lampu", "light")  
 noun("baja", "fertilizer")  
 noun("nasi", "rice")  
 noun("ikan", "fish")  
 noun("lembu", "cow")  
 noun("telur", "egg")  
 noun("nama", "name")  
 noun("tahun", "year")  
 noun("bulan", "month")  
 noun("hari", "day")  
 noun("tarikh", "date")  
 noun("baju", "shirt")  
 noun("seluar", "trousers")  
 noun("kasut", "shoes")  
 noun("stokin", "socks")  
 noun("pensil", "pencil")  
 noun("hantu", "ghost")  
 noun("tandas", "toilet")  
 noun("jalan", "way")  
 noun("bilik", "room")  
 noun("katil", "bed")  
 noun("kerusi", "chair")  
 noun("meja", "table")  
 noun("sekolah", "school")  
 noun("kerja", "work")  
 noun("banduan", "prisoner")  
 noun("panjara", "prison")  
 noun1("lelakilelaki", "men")  
 noun("keizinan", "permission")  
 noun("perempuan", "woman")  
 noun("malam ini", "tonight")



noun("badminton", "badminton")  
 noun("pertunjukan", "the show")  
 noun("pertunjukan", "show")  
 noun("cuti", "vacation")  
 noun("minggu", "week")  
 noun("alamat", "address")  
 noun("arkitek", "architect")  
 noun("dokter", "doctor")  
 noun("guru", "teacher")  
 noun("petani", "farmer")  
 noun("askar", "soldier")  
 noun("pensyarah", "lecturer")  
 noun3("Idris")  
 noun3("Zaiton")  
 noun3("Ali")  
 noun3("Malaysia")  
 noun3("Indonesia")  
 noun3("Brunei")  
 noun3("Thailand")  
 noun3("Sarawak")  
 noun3("Sabah")  
 noun3("Selangor")  
 noun3("radio")  
 prep("selepas", "after")  
 prep("antara", "between")  
 prep("oleh", "by")  
 prep("ke", "to")  
 prep("dengan", "to")  
 prep("sampai", "until")  
 prep("dari", "from")  
 prep("untuk", "for")  
 verb("dapat", "get")  
 verb("hendak", "want")  
 verb("mengambil", "take")  
 verb("diambil", "taken")  
 verb("tinggal", "live")  
 verb("jemput", "invite")  
 verb("datang", "come")  
 verb("telah mendapat", "got")  
 verb("tambah", "plus")  
 verb("buang", "through")  
 verb("jual", "sell")  
 verb("beli", "buy")  
 verb("bayar", "pay")  
 verb("belasah", "beat")  
 verb("gigit", "bite")

verb("pukul", "hit")  
 verb("bercakap", "talking")  
 verb("bunuh", "kill")  
 verb("ganggu", "disturb")  
 verb("kembali", "return")  
 verb("pergi", "go")  
 verb("pergi", "going")  
 verb(" ", "do")  
 verb("buat", "doing")  
 verb("bakar", "burn")  
 verb("baca", "read")  
 verb("anggap", "assume")  
 verb("makan", "eat")  
 verb("makan", "eats")  
 verb("bantu", "help")  
 verb("bantu", "helps")  
 verb("dimakan", "eaten")  
 verb("akan", "will")  
 verb("tunggu", "wait")  
 verb("tunggu", "waits")  
 verb("suka", "like")  
 verb("buat", "doing")  
 verb("suka", "likes")  
 verb("main", "play")  
 verb("main", "plays")  
 verb("datang", "comes")  
 verb("berjalan", "walk")  
 verb("berjalan", "walks")  
 verb("telah", "had")  
 verb("mencintai", "love")  
 verb("mencintai", "loves")  
 verb("mencuri", "stolen")  
 verb1("mempunyai", "has")  
 verb1(" ", "be")  
 verb1(" ", "been")  
 verb1("adalah", "is")  
 verb1("adakah", "has")  
 verb2("adakah", "have")  
 verb2("mempunyai", "have")  
 verb2("adalah", "are")  
 verb3(" ", "was")  
 verb3(" ", "were")  
 verb4("balik", "went")  
 verb4("lagi", "again")  
 verb4("siapkan", "finished")  
 verb4("melawat", "visited")



verb5("lalu", "passing")  
 verb5("menyanyi", "singing")  
 verb5("menari", "dancing")  
 adj("silap", "wrong")  
 adj("betul", "correct")  
 adj("kedekut", "stingy")  
 adj("lepas", "last")  
 adj("jahat", "bad")  
 adj("bagus", "nice")  
 adj("bahaya", "dangerous")  
 adj("bodoh", "stupid")  
 adj("bijak", "intelligent")  
 adj("awal", "early")  
 adj("lambat", "late")  
 adj("manis", "sweet")  
 adj("masam", "sour")  
 adj("panas", "hot")  
 adj("sejuk", "cold")  
 adj("rosak", "spoil")  
 adj("aktif", "active")  
 adj("akhir", "final")  
 adj("adil", "fair")  
 adj("panjang", "long")  
 adj("pendek", "short")  
 adj("tinggi", "tall")  
 adj("besar", "big")  
 adj("menang", "won")  
 adj("kecil", "small")  
 adj("cantik", "beautiful")  
 adj("kacak", "handsome")  
 adj("comel", "cute")  
 adj("hodoh", "ugly")  
 adj("baik", "good")  
 adj("licin", "smooth")  
 adj("rajin", "hardworking")  
 adj("aman", "peaceful")  
 adj("malas", "lazy")  
 adj("sibuk", "busy")  
 adj("malu", "shy")  
 adj("biru", "blue")  
 adj("merah", "red")  
 adj("kuning", "yellow")  
 adj("hitam", "black")  
 adj("putih", "white")  
 punct(".")  
 punct("!")

punct("?")  
punct(",")

## APENDIX C (MANUAL PENGGUNA)

### APAKAH PENGGUNAAN SISTEM INI

Sistem ini merupakan salah satu penterjemahan teks daripada Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya. Sistem ini dibangunkan dengan menggunakan perisian Visual Prolog 5.2. Semasa menjalankan penterjemahan, sistem ini mampu melaksanakan tugas seperti berikut:-

- Dapat menterjemah Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya
- Dapat menentukan tarabahasa yang betul, contohnya yang berbeza dengan tense
- Dapat menentukan kepunyaan (possession) dalam Bahasa Inggeris dengan betul
- Dapat menentukan tense ambiguity lalu satu perkataan yang berkemungkinan mempunyai lebih daripada satu makna
- Dapat menentukan struktur ayat yang betul

### CARA PEMASANGAN

Sebelum menggunakan sistem penterjemahan ini, pemasangan perlu dilakukan terlebih dahulu. Ia dapat dilakukan seperti langkah-langkah berikut:-

1. Salin (copy) folder yang bernama Translator system ke dalam direktori pilihan anda pada komputer peribadi anda.
2. Buka folder Translator system, buka fail Exe, klik gambar yang bernama Translate (X:Translator system Exe\Translate)
3. Bagi memudahkan anda, buat shortcut icon bagi Translate ke desktop
4. Anda boleh menggunakan sistem penterjemah ini.



## APENDIKS C (MANUAL PENGGUNA)

### APAKAH PENGGUNAAN SISTEM INI

Sistem ini merupakan sebuah alat penterjemahan teks daripada Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya. Sistem ini dibangunkan dengan menggunakan perisian Visual Prolog 5.2. Semasa menjalankan penterjemahan, sistem ini mampu melaksanakan tugas seperti berikut:-

- Dapat menterjemah Bahasa Melayu kepada Bahasa Inggeris dan sebaliknya
- Dapat menentukan tatabahasa yang betul, contohnya yang melibatkan *tense*
- Dapat menentukan kepunyaan (possession) dalam Bahasa Inggeris dengan betul
- Dapat menentukann *sense ambiguity* iaitu suatu perkataan yang berkemungkinan mempunyai lebih daripada satu makna.
- Dapat menentukan struktur ayat yang betul

### CARA PEMASANGAN

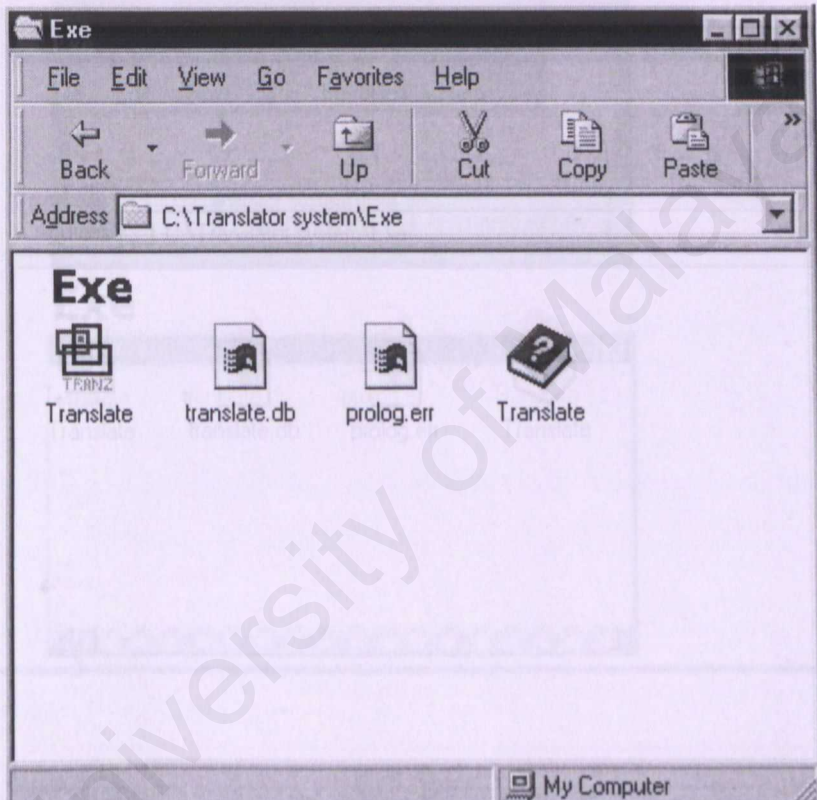
Sebelum menggunakan alat penterjemahan ini, pemasangan perlu dilakukan terlebih dahulu. Ia dapat dilakukan seperti langkah-langkah berikut:-

1. Salin (copy) folder yang bernama Translator system ke dalam direktori pilihan anda pada komputer peribadi anda.
2. Buka folder Translator system, buka fail Exe, klik gambar yang bernama Translate.( X:\Translator system\Exe\Translate)
3. Bagi memudahkan anda, buat shortcut icon bagi Translate ke desktop
4. Anda boleh menggunakan sistem penterjemah ini.

## CARA PENGGUNAAN SISTEM PENTERJEMAH INI

Untuk menggunakan sistem ini adalah mudah, hanya dengan mengikuti langkah-langkah berikut:-

1. Klik pada ikon yang bernama Translate

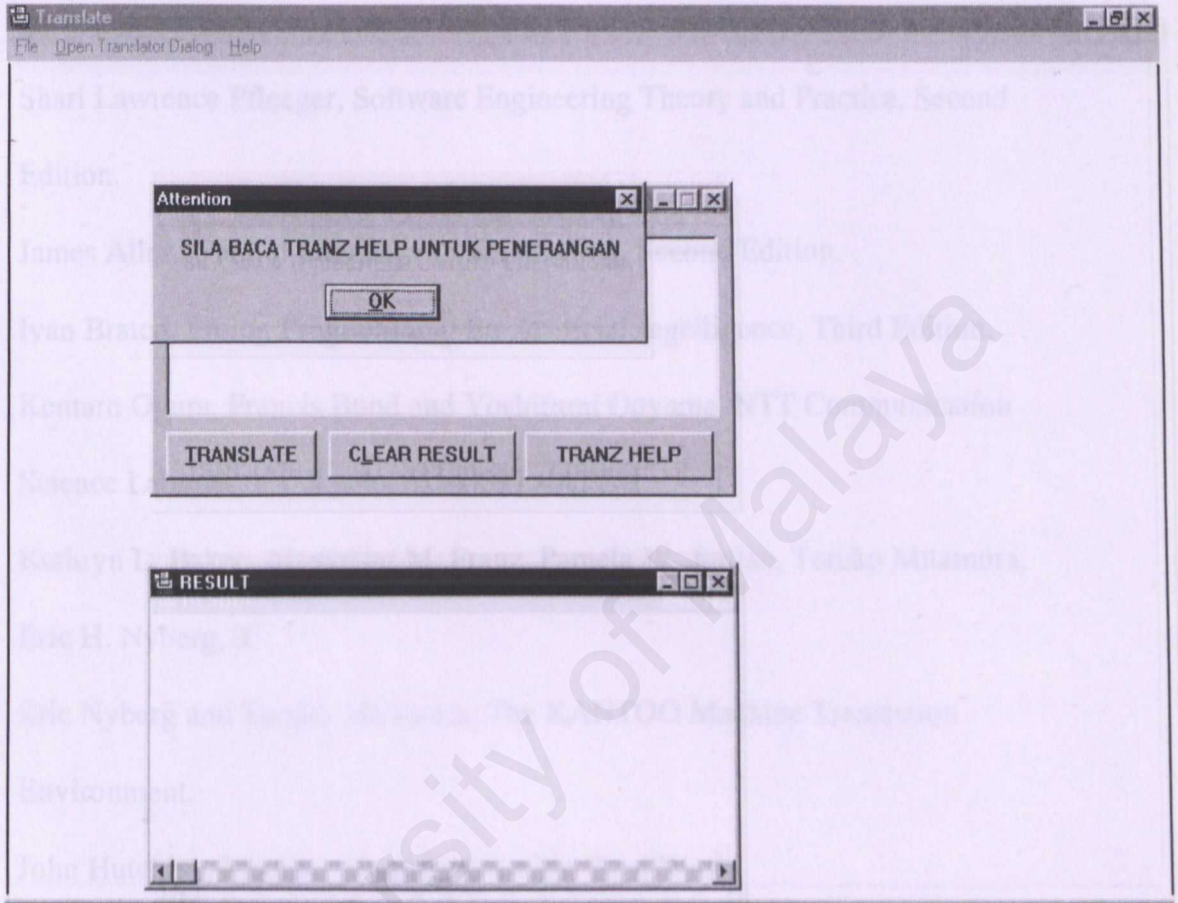


2. Window Translate akan dipaparkan

3. Klik menu Open Translator Dialog.



4. Window Translate serta dialog amaran akan dipaparkan dan klik ok.



5. Masukkan teks yang hendak diterjemahkan
6. klik TRANSLATE untuk melihat hasil terjemahan. Hasil terjemahan akan dipaparkan pada window RESULT
7. klik CLEAR RESULT untuk memadamkan hasil terjemahan pada window Result
8. Untuk keluar dari sistem ini, klik palang pada sudut kanan atas pada window Translate

## BIBLIOGRAFI

1. George F. Luger & William A. Stubblefield, Artificial Intelligence ,Structures and Strategies for Complex Problem Solving, Third Edition.
2. Shari Lawrence Pfleeger, Software Engineering Theory and Practice, Second Edition.
3. James Allen, Natural Language Understanding, Second Edition.
4. Ivan Bratco, Prolog Programming for Artificial Ingelligence, Third Edition.
5. Kentaro Ogura, Francis Bond and Yoshifumi Ooyama, NTT Communication Science Laboratories, Kyoto, 619-0237, JAPAN.
6. Kathryn L. Baker, Alexander M. Franz, Pamela W. Jordan, Teruko Mitamura, Eric H. Nyberg, 3<sup>rd</sup>
7. Eric Nyberg and Teruko Mitamura, The KANTOO Machine Translation Environment.
8. John Hutchins, Machine Translation in The Real World
9. <http://utmk-ultra.cs.usm.my>
10. <http://www.systransoft.com>
11. [www.epedoman.com](http://www.epedoman.com)
12. <http://sangenjaya.arc.net.my>
13. [www.ewgate.com](http://www.ewgate.com)